On the Orchestration of Federated Learning Through Vehicular Knowledge Networking

Duncan Deveaux[†], Takamasa Higuchi[‡], Seyhan Uçar[‡], Chang-Heng Wang[‡], Jérôme Härri[†], Onur Altintas[‡]

[†]EURECOM, Campus SophiaTech, 450 route des Chappes, 06904 Sophia-Antipolis, France
E-mail: {deveaux,haerri}@eurecom.fr
[‡]InfoTech Labs, Toyota Motor North America R&D, Mountain View CA, USA
E-mail: {takamasa.higuchi,seyhan.ucar,chang-heng.wang,onur.altintas}@toyota.com

***Abstract*—Federated Learning (FL) is a recent distributed technique to extract knowledge, i.e. an abstract understanding obtained from a set of information through experience and analysis. Vehicular networks are highly mobile networks in which a large spectrum of data types is distributed. So far, no mechanisms have been defined that distribute FL model updates in vehicular networks based on which nodes are likely to hold the right data for training, and when. In turn, this potentially limits FL model training speed and accuracy. In this paper, we describe protocols to exchange model-based training requirements based on the Vehicular Knowledge Networking framework. Based on this understanding, we define vehicular mobility and data distribution-aware FL orchestration mechanisms. An evaluation of the approach using a federated variant of the MNIST dataset shows training speed and model accuracy improvements compared to traditional FL training approaches.**

***Index Terms*—vehicular, federated, knowledge, orchestration**

I. Introduction

Knowledge plays an increasingly important role in vehicular networks. In applications such as platooning, object recognition or navigation, knowledge extracted from sensor information can be generated, stored and shared to augment the capacities of connected vehicles. Machine learning has increasingly been used as a tool for generating knowledge, as part of multiple use cases. Traditionally, model training is performed by a single, centralized organization after all training information has been gathered in a single point. This mode of operation potentially raises privacy and bandwidth issues as large amounts of training information are transferred over the network to third-party training servers. Federated Learning (FL) is a novel machine learning paradigm aimed at addressing these issues. Instead of gathering all training information in a single point, model training is directly performed by data-collecting nodes. The result of each local training is then aggregated.

Mechanisms have been studied that optimize FL training in vehicular networks based on heterogeneous computing powers and link quality of vehicles. However, vehicular networks are highly mobile and involve large amounts of information of different nature. Vehicular mobility introduces a specific set of challenges for the efficient application of FL in vehicular networks. Because of mobility, the nature and amount of available training information on-board vehicles varies dynamically. What is more, as vehicles are bound by predefined and nonrandom mobility patterns, mobility induces a biased distribution of training data among vehicles.

As such, a vehicular mobility-based orchestration mechanism for FL is required to improve training performances. In this article, we suggest an approach of mobility-based FL orchestration based on *Vehicular Knowledge Networking* (VKN), a framework for knowledge description, creation, storage and distribution in vehicular networks. Through two complementary use case studies, we demonstrate the need for and define a mobility-based FL orchestration mechanism. The contribution of this paper lies in FL orchestration through training client selection in vehicular environments where training input is unevenly distributed, rather than in FL algorithms themselves. In the evaluation section, we show performance improvements of VKN-orchestrated FL while training a classifier model for the digits-only FEMNIST dataset [1]. It is a federated version of the MNIST dataset of handwritten digits, grouped by writer.

The rest of the article is organized as follows: Section II introduces the concept of FL, in both centralized and decentralized settings. Section III introduces the challenges that mobility introduces for vehicular FL through two use cases. Based on this understanding, Section IV describes the need for a vehicular mobility-based orchestration of FL. Section V introduces *Vehicular Knowledge Networking* (VKN) as a potential technical candidate to realize FL orchestration. Finally, Section VI shows performance improvements linked to VKN-orchestrated FL, while Section VIII summarizes the article.

II. Federated Learning

A traditional approach to training a machine learning model is to upload all training information from multiple local sources to a centralized training organization. A learning algorithm, e.g. Stochastic Gradient Descent (SGD), is then applied to the full input set to train a model. Once fully trained, the model is spread to local nodes so it can be used in vehicular applications. However, this approach notably raises privacy issues, as it requires exposing the training information from each local source to a central authority.

FL has emerged in the past years to address the issue as an alternative approach to traditional machine learning. Instead of gathering information from local nodes in a centralized

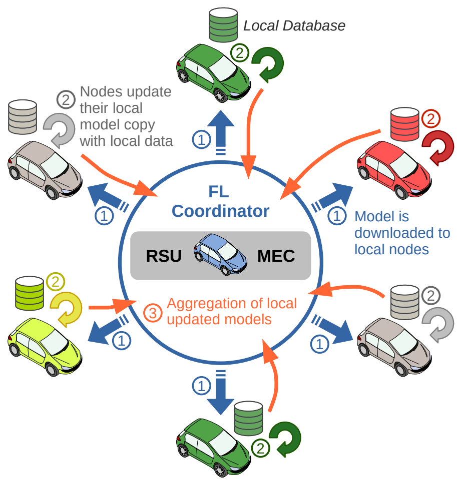


Fig. 1: A Round of Federated Learning Training

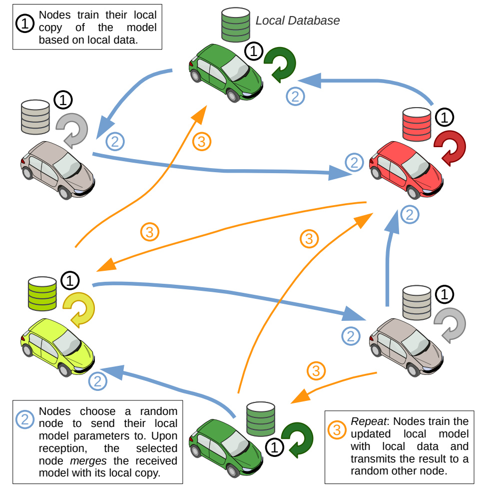


Fig. 2: Principles of Gossip Learning

data center for subsequent training, FL training is directly performed by local nodes using local information.

As illustrated in Figure 1, training a FL model is decomposed into training steps:

1. A subset $s \in S$ of local nodes is selected to perform local training.
2. The current global model m_{glob} hosted by the coordinator is downloaded to each node in s (see Figure 1, step 1).
3. Each node $s_i \in S$ performs one or several steps of training of the received model with local information (see Figure 1, step 2).
4. The parameters of each locally-updated model are sent back to the coordinator, which aggregates the local model parameters to update m_{glob} (see Figure 1, step 3).

The FL approach allows training a model from distributed input information while preserving user privacy. Alternatively, decentralized approaches eliminating the FL coordinator such as Gossip Learning (GL) have been studied in the literature [2]. As shown in Figure 2, during a session of GL training, each node implements two procedures:

- *Local training and distribution:* Each local node i runs a few steps of local training of their local copy of the model m_i. The updated local model parameters are then sent to a random other node (see Figure 1, step 1).
- *Local model merging:* When receiving a model update, a local node merges the received parameters with those of its own copy of the model (see Figure 1, step 2). Then, it repeats a step of local training and distribution (step 3).

After multiple rounds of local training and local model distribution to random neighbors, models in the network converge.

III. Vehicular Mobility Challenges for FL

Theoretically, FL/GL allows comparable training performance to that of traditional machine learning, while ensuring stronger training information privacy [3]. However, practical challenges must be addressed in order to reach such performance in real mobile applications. Typical challenges faced in applications of FL to mobile networks include:

- Resource constraints [4, 5]
  - Non-perfect network channel quality
    - Link quality-aware training node selection [6].
    - Irrelevant updates discarding mechanism [7].
    - Compression mechanisms [8, 9].
  - Computational power.
  - Access to training input.
    - Heterogeneous sensing capabilities.
    - Limited storage capacity.
- Non independent and identically distributed (IID) training input.
- Security concerns
  - Protection against forged model updates [10].
  - Privacy of training input [11].
- Optimal convergence of the distributed gradient descent algorithm [12, 13].
- Incentive mechanisms for FL model training [14, 15].

When training a FL/GL model, each node is associated with a training context. An uneven distribution and access to training data, as well as nodes' training ability, linked to computing power and network link quality must be taken into account. A key feature of vehicular networks is the high mobility of vehicles. As introduced in [16], vehicular mobility brings specific challenges for FL training. It implies a dynamically evolving training environment and distribution of training data among nodes. In this section, we exemplify the vehicular mobility-related challenges for FL training in vehicular networks through two complementary case studies.

A. Case Study 1: Commonly Observable Input

We consider the use case of a FL model that can be trained from input that can be observed by vehicles at all times while driving. Vehicles sense training input as they drive using on board sensors and are then connected with a FL coordinator

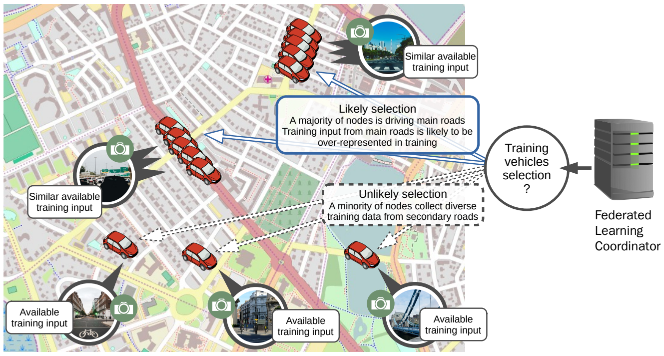


Fig. 3: Vehicular Mobility Impact for FL with Commonly Observable Data

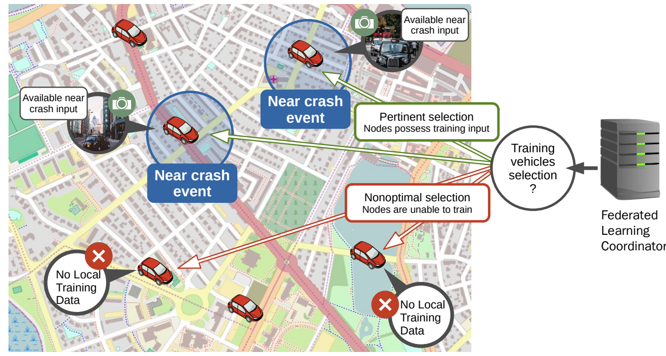


Fig. 4: Vehicular Mobility Impact for FL with Sparsely Observable Data

to train a model from their locally observed data. In that case, the trained model uses input data that can be collected at any time of driving by vehicles, e.g. road-state related information.

For example, [17] describes a machine learning model of road roughness estimation from vehicles' front camera. Although the authors train the model using traditional machine learning, the model can be adapted for FL training, where sensed training information is kept on board vehicles.

At each step of FL training, the coordinator performs client selection and chooses a set of vehicles to locally compute a model update. The nature of mobility in vehicular networks brings additional challenges to perform efficient FL client selection. As illustrated by Figure 3, mobility in vehicular networks is non random. Two originally distant vehicles are likely to drive the same roads at some point in their journey, as they drive through main arteries e.g. in cities.

As such, the unequal distribution of vehicles on the road network leads to many vehicles sensing similar training input in similar areas – e.g. camera images of main roads. A mechanism must be designed to unsure that nodes possessing diverse training data sensed in secondary roads also get represented in FL training, in order to avoid over fitting the model to typical main axes mobility.

B. *Case Study 2: Sparsely Observable Input*

We consider the use case of a FL model that can be trained from input that can only be observed by vehicles at punctual times and locations. For example, [18] presents a machine learning model to predict near-crash situations from observed vehicles' kinematics. Near-crashes are reasonably rare events, that can only be observed at punctual points of space and time. In the hypothesis of training this model using FL, only vehicles having sensed kinematics data of another vehicle experiencing a near-crash should be leveraged as local training nodes. As vehicles are moving, the ability of vehicles to take part in training dynamically evolves through time.

In order to accurately train the model, the selected clients should be nodes having witnessed a near-crash event and sensed the accompanying kinematics information. As illustrated by Figure 4, an orchestration mechanism must be defined in order to allow the coordinator and training vehicles to communicate training capacities and allocate training resources to vehicles that hold meaningful input.

IV. A Need for Mobility-based FL Orchestration

Various orchestration mechanisms have been defined to increase the efficiency of FL in mobile networks to account for heterogeneous capacity and resources access in the network. For example, [6] shows performance improvement of FL in mobile networks by favoring the selection of training nodes with good channel link quality.

In the previous section, we exemplified the vehicular mobility-related challenges inherent to the application of FL in vehicular networks. In this work, we aim at defining FL orchestration mechanisms to address challenges related to vehicular mobility, leading to varying training environments, rather than varying training capabilities of vehicles.

A. *Requirements for Mobility-based Orchestration*

Mechanisms of mobility-based orchestration of FL in vehicular networks must account for the dynamically evolving training environment of vehicles, in order to:

- Increase the likelihood that nodes selected for model training are able to train the model – i.e. possess the right input for the right model.
- Ensure a dynamic and balanced selection of vehicles for model training. It should ensure to (i) avoid over fitting the trained model with training input sensed in similar contexts, and (ii) avoid missing valuable and original training input sourced from some vehicles.

Protocols should be defined so that training nodes and coordinators can communicate about required training environments to train a given model.

On the one hand, training nodes e.g. vehicles should communicate information about the nature of training data they hold and the context in which it was gathered. Without leaking training data, a vehicle should be able to indicate what category of information it possesses through semantics and information naming standards.

On the other hand, coordinators must indicate the necessary context and necessary nature of input training data needed for participating in a FL model training, so as to avoid contacting vehicles that lack required training input.

Once the coordinator was provided with contextual information about the current training context vehicles and training nodes are facing, mechanisms should be defined to dynamically allocate client selection to avoid over representing or missing valuable training input.

B. Technical Definitions

In order to allow communication of contextual knowledge between training nodes, the following technical requirements must be met:

1. A message format should be defined to describe the nature of information sensed by vehicles, and the context it is sensed in. This message should include at least:
   - A standard, semantically-defined name to identify the nature of sensed information.
   - The time and position of information collection.

   Additional pertinent contextual information includes (i) the status of the vehicle: driving, idle, (ii) data on vehicle kinematics and destination.
2. A message format should be defined to formally describe FL models and their required input. Through a formal definition of the conditions of application of a model, coordinators can perform efficient training node selection by targeting the nodes that possess the required training input in the right format. Elements to formally describe a model should at least include input and output information types and nature.
3. A message format should be defined for coordinators to request for vehicles that possess the right context for training a model.

V. Integration with Vehicular Knowledge Networking

Vehicular Knowledge Networking (VKN) is a framework for knowledge description, creation, storage and distribution within vehicular networks [19]. VKN aims at providing a framework for standard knowledge description, in turn allowing cooperative and interoperable knowledge creation and distribution. As such, VKN can provide for the technical requirements of mobility-based orchestration of FL in vehicular networks.

As illustrated by Figure 5, vehicular mobility-based orchestration of FL requires the three aspects of VKN: knowledge description, distribution and storage.

A. VKN Semantics & Input Description

VKN allows the description of a model's input and output parameters. As defined in [19], the use of semantics standard allows vehicles to know which input to provide to a model in order to apply or train it. Using this format, training nodes and coordinator can reach a common understanding on what context is required to train a model.

Figure 6 illustrates an example model description applied to the model of road surface condition estimation described

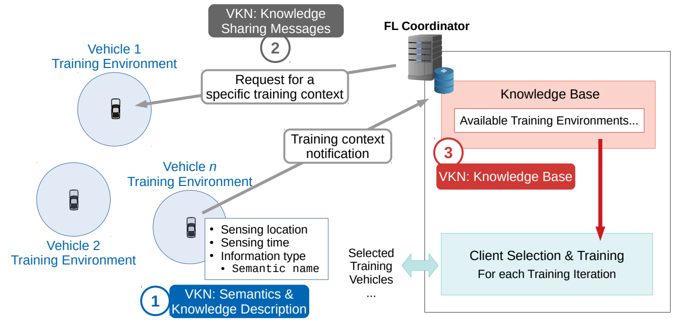


Fig. 5: FL Orchestration Through VKN

in [17] and introduced in Case Study 1. The nature of required input to the model is a front camera image, the output of the model is a class for the road surface condition, as detailed in [17]. Additionally, description of model weights as well as training parameters allow nodes to take part in a training process for the model.

```
model_name: model.road_surface_condition

model_description:
    INPUT(camera_image: #Image.FrontCamera),
    OUTPUT(road_surface_condition: #RSC.ClassID)

model_training_parameters:
    MODEL_PARAMETERS(... [Neural Network
                    Weights Description]),
    LOSS_FUNCTION(loss.cross_entropy)
```

Fig. 6: Example of a VKN Model Description applied to [17] (Case Study 1)

B. VKN Knowledge Distribution: Environment Description

Coordinator and training vehicles must communicate in order to orchestrate FL training:

- Vehicles communicate their training environment to the coordinator.
- The coordinator issues training environment requests to vehicles to get a list of vehicles able to train a model.

As exemplified in pseudo code in Figure 7, using the Vehicular Knowledge Query format introduced in [19], coordinator and vehicles can communicate on training environments. The description of a training environment includes the nature of sensed information – i.e. a semantically-defined name –, as well as the time and location of information collection. In a decentralized GL setting, vehicles can directly query and exchange training environments with other vehicles.

C. VKN Knowledge Base: Coordinator

Finally, as shown in point 3 of Figure 5, the coordinator receives training environment descriptions from vehicles and stores them in a local Knowledge Base (KB). Using the content of the KB, while training a model, the coordinator is able to select vehicles that are most likely to feature the required training input. Alternatively, in a decentralized GL setting where

```
Coordinator requests:
 REQUEST TRAINING_ENV(#Image.FrontCamera)
         IN [Area] AT [Time Range]

Vehicle answers:
 AVAILABLE TRAINING_ENV(#Image.FrontCamera)
           IN [Area] AT [Time Range],
     VEHICLE ID: [vehicle address]
```

Fig. 7: Training Environment Request and Answer, based on [17] (Case Study 1)

vehicles directly exchange training environments, vehicles can store that content in an on-board KB and use it to directly transfer model updates to other pertinent vehicles.

VKN integration provides a support for the definition of mobility-based FL orchestration mechanisms in vehicular networks. In Section VI, we provide and evaluate a VKN-based orchestration mechanism for centralized FL. Vehicles and coordinator are training a model whose VKN description is commonly available. As vehicles sense information that qualifies as input for the model being trained, they inform the coordinator of their updated training environment and ability to train the model. Finally, at each iteration of model training, the coordinator performs clustering of potential training vehicles based on the location of their last reported training input sensing. Then, for each cluster, the coordinator selects the vehicle having most recently reported an updated training environment to perform a model update for that iteration.

VI. Evaluation Setup

We implement mobility-based orchestration of FL through VKN and investigate potential performance improvements for both case studies described in Section III. We simulate a set of vehicles in a fixed size area, whose movements follow standard mobility models. As they move, vehicles sense information about their environment, that can be used as FL training input. We use vehicles as training nodes to train a classifier based on the digits-only FEMNIST dataset.

For the first case study, we investigate on mobility-based orchestration on FL training process for models with commonly observable input, e.g. the road surface condition estimation model in [17]. All vehicles periodically sense information from their environment that is usable in the model's FL training. As vehicular mobility is not random, two vehicles that are originally distant are likely to drive the same road at some point, and thus provide similar input to the model's training, potentially triggering overfitting.

For the second case study, we implement mobility-based orchestration for the class of models that can be trained on input sourced from sparse, rare events. The model described in [18] fits this category. Due to vehicular mobility, in this case study, only a fraction of vehicles owns the required input for training the model at a given time.

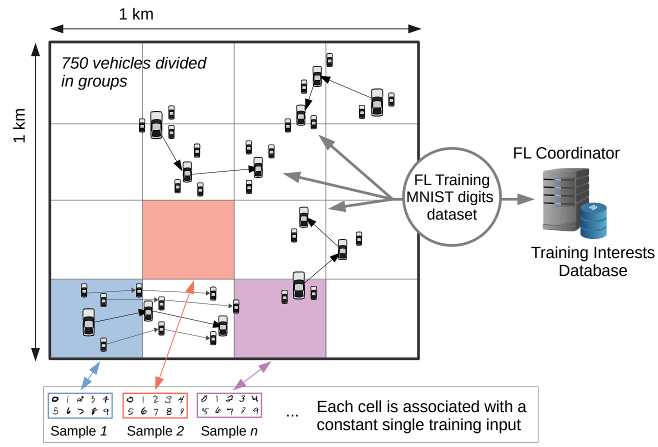


Fig. 8: Evaluation Setup for Case Study 1

A. MNIST Dataset Choice

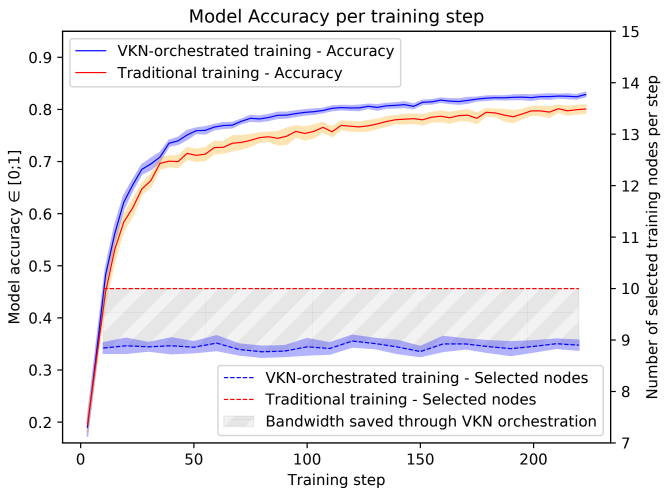
In order to assess the performance of FL training with VKN orchestration, we train a classifier based on the digits-only FEMNIST dataset [1]. It is a federated version of MNIST, and consists of a dataset of handwritten digits associated with a text version of the character. In this federated version, digits are grouped by writer. As such, digits-only FEMNIST replicates the non-IID data distribution that one should expect in FL training.

Rather than in FL algorithms themselves, the contribution of this paper lies in FL orchestration through training client selection in vehicular environments. While digits-only FEMNIST is a simple dataset, it provides flexibility in terms of choosing and modifying the distribution of training data, which is necessary to verify the performance of the VKN-orchestrated FL scheme. Therefore, as an initial application and proof of concept, we use the FEMNIST dataset to compare the performance of training with and without VKN orchestration.

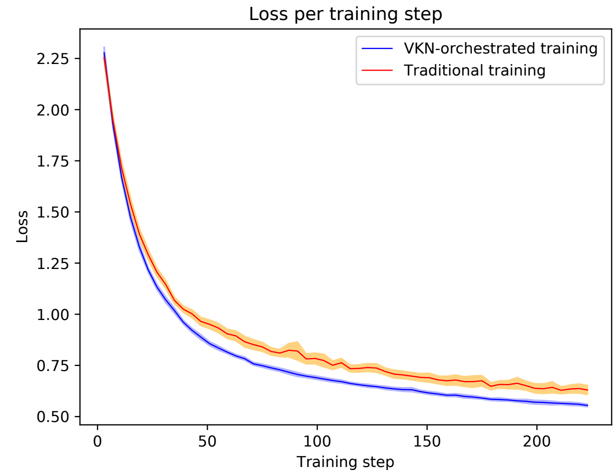
B. General Setup

We simulate the FL training of a digits-only FEMNIST classifier in a vehicular environment. $V = 750$ vehicles are simulated in a one square kilometer area over the course of $T = 1h$ simulated time. Each vehicle is a potential training node, reachable by a central coordinator at all times to compute local model updates upon request. For both case studies, the training algorithm and evaluation processes are identical. However, the mobility model followed by vehicles as well as the data sensing mechanism are different, to simulate the mobility challenges described in Section III. We use the *tensorflow federated* library to implement the model training through the *Federated Averaging* algorithm, which is an implementation of FL.

1) Mobility & Data sensing: Vehicles move according to a standard mobility model that is specific to each of the two case studies. The first simulation uses the Random Waypoint (RWP) mobility model, whereas the second uses the Reference Point Group (RPG) mobility model. Moreover, training input



(a) Model Accuracy per Training Iteration



(b) Training Loss per Iteration

Fig. 9: Evaluation of VKN-orchestrated FL Performances for Case Study 1

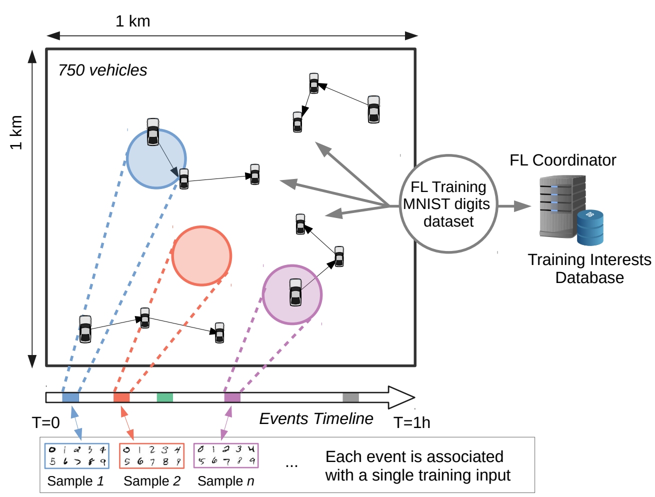


Fig. 10: Evaluation Setup for Case Study 2

sensing mechanisms are simulated differently in both cases, respectively a grid-based observation mechanism for the first case study and a rare events generation mechanism for the second.

2) Model Training: The FL coordinator holds the current global state of the FEMNIST-based model to be trained. The training is divided into sequential steps in which:

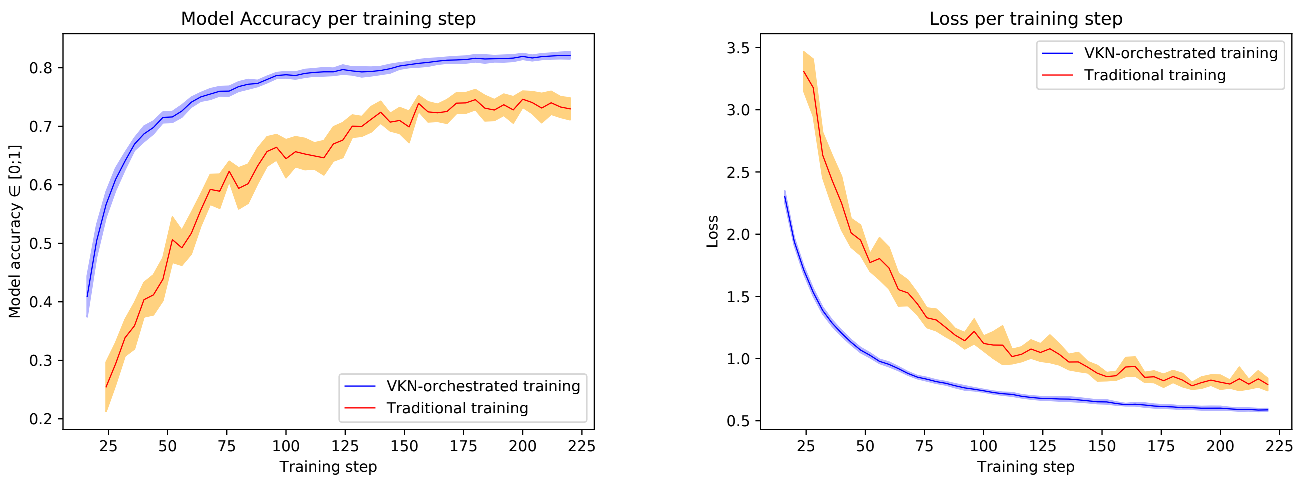
1. The coordinator samples a set of $N = 10$ vehicles to which it transmits the current global state of the model.
2. Upon receiving the model:
   - If the vehicle holds training input for the model in its local storage, a model update is locally computed and returned to the coordinator.
   - If not, the vehicle is not able to train the model due to a lack of training data, and discards the model.
3. After a simulated delay of $\Delta_t = 15s$, the coordinator aggregates the model updates received for the current step. It then starts a new step and repeats until convergence.

3) VKN Orchestration: In both simulations, we compare the efficiency of 'traditional' FL as opposed to mobility-based VKN-orchestrated FL by analyzing model accuracy and training loss per training iteration. The VKN orchestration process takes place at the training client selection level in the coordinator. Thus, the difference between the *traditional* and *VKN-orchestrated* approaches lies in the training vehicle sampling mechanism at each step.

On the one hand, in the traditional approach, no mechanisms are implemented to anticipate which vehicle is likely to hold which type of training information. At every training step, the coordinator samples $N = 10$ randomly chosen vehicles to perform local model training.

On the other hand, in the VKN-orchestrated approach, vehicles inform the coordinator of their training environment respectively (i) every $\Delta_t = 10s$ in case study 1 where the trained model requires commonly observable input, and (ii) each time a relevant input is sensed in case study 2 where the trained model requires rarely sensed input. Thus, the coordinator is fitted with a KB containing the last observed pertinent input of each vehicle if any, as well as the time and location of sensing. Training environment information is discarded from the coordinator's KB after a timeout of $\Delta_t = 5min$.

Using this contextual knowledge, the coordinator clusters the training vehicles by the location of their last documented training input sensing in $N = 10$ clusters using a K-Means algorithm. Then, the coordinator sends a model update to the vehicle having most recently filed a training environment item in each of the N clusters. Through that clustering process, the coordinator maximizes the diversity of training input, while reducing the likelihood to request training to vehicles that do not possess the required input data.



(a) Model Accuracy per Training Iteration

(b) Training Loss per Iteration

Fig. 11: Evaluation of VKN-orchestrated FL Performances for Case Study 2

VII. Performance Evaluation

In this section, we describe the specific parameters of the simulations performed to represent each case study, and discuss evaluation results.

A. Case Study 1: Commonly observable input

Figure 8 illustrates the specific setup of Simulation 1. The vehicles in the simulation follow the RPG mobility model. Mobility is divided into one major and nine minor groups of vehicles. In order to represent commonly observable training input, the simulation area is divided into a grid of 10x10 cells. Each cell is permanently associated with a single, constant training sample extracted from the digits-only FEMNIST dataset. Vehicles periodically sense the training sample associated with the cell they are located in, every $T = 10s$ of simulated time.

As described in the evaluation's general setup, at each step, the coordinator performs clustering of potential training vehicles in order to select $N = 10$ clients most likely to feature diverse training input. In this case study, as an optimization, if two selected vehicles are separated by a distance of 50 meters or less, one of them is discarded from the selection. Thus, in the VKN-orchestrated approach, for some training iterations, the number of selected vehicles can potentially be below $N = 10$.

Figure 9 shows the obtained result in terms of model accuracy and training loss per training iteration. The results are bounded by 95% confidence intervals obtained after averaging the results from $S = 25$ simulations. In this case study, orchestration resulted in: (i) a minor but consistent improvements in terms of model accuracy of about 4% while (ii) reducing the amount of nodes selected for training to an average of 9 instead of 10 training vehicles per iteration. Thus, VKN orchestration combines improved training accuracy with reduced bandwidth usage in terms of model update exchanges. As a trade off, vehicles inform the coordinator of their training environment every $\Delta_t = 10s$. Thus, the actual variation of bandwidth usage depends on the trained model and the difference between the size of model updates and that of training environment messages. In complex neural networks models encoding numerous weights, the bandwidth saved through the reduction of model update exchanges is likely to be significant over the overhead of training environment messages exchange.

B. Case Study 2: Sparsely Observable Input

Figure 10 illustrates the specific setup of Simulation 2. Vehicles in the simulation follow the RWP mobility model, with a pause probability of 0.5, a maximal pause duration of 5 minutes, and a speed range of $[5; 20]m/s$. As the simulation aims to represent situations of sparsely observable training input, events are randomly generated within the simulation area.

Events occur following a Poisson distribution at a constant mean rate of 1.5 events per simulated minute. Each event has a circular shape and appears in the simulated area for a limited amount of time. Namely, an event is composed of (i) a start time, (ii) a duration sampled from a normal law $\mathcal{N}(60, 10)$ seconds, (iii) a center point sampled uniformly in the simulated area, (iv) a radius $r \sim \mathcal{N}(50, 1)$ meters, and (v) a single training data sample extracted from the digits-only FEMNIST dataset that passing vehicles will be able to sense.

Everytime a vehicle drives within the bounds of the circular representation of an active event, the vehicle senses the training data sample associated with the event and saves it in local storage, erasing any previous sensing. Moreover, to account for existing vehicular database maintenance features [20], data samples stored on-board vehicles are removed if:

Time relevance The last sensed sample was obtained more that 5min of simulated time ago.

Space relevance The last sensed sample was obtained more than 500m away.

Figure 11 illustrates the obtained results in terms of model accuracy and training loss per training step. The results are bounded by 95% confidence intervals obtained after averaging the results from $S = 25$ simulations. The orchestration of the FL training through VKN results in significant improvements in terms of convergence time as well as model accuracy. After convergence, the model training through VKN orchestration shows a significance increase in label prediction accuracy of about 15%.

VIII. Conclusion

Federated Learning shows potential for knowledge creation and distribution in vehicular networks, while avoiding the transfer of training data. However, the patterns of vehicular mobility introduce specific challenges for FL, such as a dynamically evolving training environments for nodes in vehicular networks. In this paper, we exemplified vehicular mobility-related challenges for FL through two complementary case studies. On the one hand, models with commonly observable input require educated training node selection in order to avoid over fitting models to a majority of input sourced from main roads axes. On the other hand, models that require rare observations as input, e.g. models studying near-crash events require communication between training nodes, to avoid requesting model updates to vehicles that do not own the required input data. We introduce *Vehicular Knowledge Networking* as a candidate framework to answer needs of mobility-based orchestration in vehicular FL. Finally, we evaluate the described approach by running two complementary simulations representing each case study. We observe that mobility-based FL orchestration through VKN led to improvements in terms of model training speed and model accuracy, as opposed to unorchestrated training.

References

[1] Sebastian Caldas et al. *LEAF: A Benchmark for Federated Settings*. 2019. arXiv: 1812.01097 [cs.LG].

[2] István Hegedűs, Gábor Danner, and Márk Jelasity. "Gossip Learning as a Decentralized Alternative to Federated Learning". In: *Distributed Applications and Interoperable Systems*. Ed. by José Pereira and Laura Ricci. Cham: Springer International Publishing, 2019, pp. 74–90.

[3] L. Giaretta and Š. Girdzijauskas. "Gossip Learning: Off the Beaten Path". In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 1117–1124.

[4] Ahmed Imteaj et al. *Federated Learning for Resource-Constrained IoT Devices: Panoramas and State-of-the-art*. 2020. arXiv: 2002.10610 [cs.LG].

[5] Wei Yang Bryan Lim et al. *Federated Learning in Mobile Edge Networks: A Comprehensive Survey*. 2019. arXiv: 1909.11875 [cs.NI].

[6] Takayuki Nishio and Ryo Yonetani. "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge". In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)* (May 2019).

[7] Luping Wang, Wei Wang, and Bo Li. "CMFL: Mitigating Communication Overhead for Federated Learning". In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 954–964.

[8] Richeng Jin, Xiaofan He, and Huaiyu Dai. *On the Design of Communication Efficient Federated Learning over Wireless Networks*. 2020. arXiv: 2004.07351 [cs.LG].

[9] Jinjin Xu et al. *Ternary Compression for Communication-Efficient Federated Learning*. 2020. arXiv: 2003.03564 [cs.LG].

[10] Eugene Bagdasaryan et al. *How To Backdoor Federated Learning*. 2018. arXiv: 1807.00459 [cs.CR].

[11] Z. Li, V. Sharma, and S. P. Mohanty. "Preserving Data Privacy via Federated Learning: Challenges and Solutions". In: *IEEE Consumer Electronics Magazine* 9.3 (2020), pp. 8–16.

[12] B. Swenson et al. "Distributed Gradient Descent: Nonconvergence to Saddle Points and the Stable-Manifold Theorem". In: *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2019, pp. 595–601.

[13] S. Wang et al. "Adaptive Federated Learning in Resource Constrained Edge Computing Systems". In: *IEEE Journal on Selected Areas in Communications* 37.6 (2019), pp. 1205–1221.

[14] Yuan Liu et al. *FedCoin: A Peer-to-Peer Payment System for Federated Learning*. 2020. arXiv: 2002.11711 [cs.CR].

[15] Han Yu et al. "A Fairness-Aware Incentive Scheme for Federated Learning". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. AIES '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 393–399. ISBN: 9781450371100.

[16] Ahmet M. Elbir, Burak Soner, and Sinem Coleri. *Federated Learning in Vehicular Networks*. 2020. arXiv: 2006.01412 [eess.SP].

[17] S. Roychowdhury et al. "Machine Learning Models for Road Surface and Friction Estimation using Front-Camera Images". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. 2018, pp. 1–8.

[18] Osama A. Osman et al. "Prediction of Near-Crashes from Observed Vehicle Kinematics using Machine Learning". In: *Transportation Research Record* 2673.12 (2019), pp. 463–473.

[19] Duncan Deveaux et al. *A Definition and Framework for Vehicular Knowledge Networking*. 2020. arXiv: 2005.14505 [cs.NI].

[20] *ETSI EN 302 895 V1.1.1 : Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)*. Sept. 2014.