Integrated Simulation of Federated Learning for Large Scale Intelligent Transportation Systems

Nagacharan Teja Tangirala*, Christoph Sommer†, and Alois Knoll*
*Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, Munich, Germany
†TU Dresden, Faculty of Computer Science, Germany
nagacharan.tangirala@tum.de, cms-labs.org/people/sommer, k@tum.de

***Abstract*—Federated Learning (FL) is a decentralized learning paradigm that is expected to play a substantial role in both the network management of 5G and beyond protocols and smart city applications like Intelligent Transportation System (ITS). The massive scale and decentralized nature of FL makes them depend on Vehicular Ad-hoc Networks (VANETs) to support model exchanges between the FL participants. However, traditional FL simulators are not designed to model the communication behavior of a VANET. We address this gap by proposing an integrated simulation approach to facilitate combined studies of FL and VANET. We describe the necessary design changes to common simulation components and implement a sample simulator by extending a simplified VANET simulator. We demonstrate the need of such integrated simulation, illustrating that typical VANET effects substantially impact FL training – and that FL training substantially impacts ITS infrastructure. We also demonstrate the fidelity and performance of our approach by comparing it with a traditional FL simulator. Finally, we discuss the applications and limitations of our proposed approach.**

***Index Terms*—Federated Learning, ITS, VANET Simulations**

I. Introduction

Federated Learning (FL), a technique for building an Artificial Intelligence (AI) model through iterative and decentralized training, is expected to play a prominent role in the network management of 5G and beyond protocols. Management of computing and network resources, load prediction, and network data analytics are some network management functions handled by FL [1].

In parallel, AI is also expected to play an integral role in enabling smart cities to achieve their sustainability goals [2] and, as an integral component of a smart city, smart mobility and Intelligent Transportation Systems (ITSs) are also affected by this push. Indeed, FL is one of the emerging paradigms that is seeing a growing adoption for ITS applications. Traffic flow prediction, traffic target recognition, route planning, and parking management are examples for FL-based ITS applications [3] – and FL could enable privacy-conscious data sharing for explainable cooperative ITS applications [4].

As FL is an AI paradigm, it depends on large amounts of data for model training. This adds additional load on the already limited communication resources. Predictions indicate that the communication requirements of the smart city will outgrow the capabilities of 5G by 2030 [5]. This, together with the massive scale and decentralized nature of FL makes them depend on Vehicular Ad-hoc Networks (VANETs) to support model exchanges between the FL participants. In addition, the hardware capabilities of the vehicles can also constrain the training process. Hence, adding FL must be thoroughly studied to make it feasible on the limited hardware and VANET resources.

Due to the large scale of both the systems under study and the required data, many researchers turn to simulators to assist in studying FL-based ITS applications [6]. Some traditional FL simulators even allow the estimation of communication costs and support methods such as quantization to reduce message size [7]. However, they are not designed to model the communication behavior of a VANET. Further, they are not suitable to model the agent-like behavior of the vehicles. Hence, there is a necessity for a simulator that supports combined modeling of FL and VANET behavior. Recognizing the need, Lobato et al. [8] proposed FLEXE for modeling FL-based ITS applications. FLEXE is designed by extending a high-fidelity VANET simulator called Veins [9]. However, the high computational complexity of Veins prohibits large-scale evaluations without the help of High-Performance Computing (HPC) resources.

In this paper, we propose an integrated simulation approach to enable studies of FL in ITS applications. Initially, we present a summary of the current research work and highlight the necessity of the simulator. Further, we describe the important adaptations of the FL behavior to consider the VANET-specific aspects. Next, we discuss the implementation choices and describe the necessary components of the simulator. We validate the implementation by comparing the results with a traditional FL simulator. Further, we demonstrate that our approach can model the combined FL and VANET behavior. Finally, we discuss the strengths and limitations of the proposed simulator and outline the potential applications.

II. Related Work

FL is applied to a variety of ITS applications. A privacy-preserving estimation of electric charging behavior is carried out using FL [10]. The vehicle charging information from multiple locations is used to train a model that forecasts the charger usage estimations for the subsequent days. A personalized zone prediction for taxi drivers is developed using FL [11]. A real-world taxi dataset is used to forecast the recommended zones for a taxi, considering individual drivers' preferences. A semi-asynchronous trajectory prediction method is trained using FL [12].

In addition to applications, improved methods of FL are proposed to make them suitable for the VANET. A reputation-based client selection scheme is proposed to account for VANET constraints [13]. A privacy-preserving aggregation method is developed for the VANET environment [14]. These methods are evaluated using virtual machines, with each client and server having dedicated computing resources. Participants communicate using inter-process communication protocols like gRPC. Scaling can be expensive for such implementations.

Alternatively, traditional FL simulators can be used for studying large-scale applications. Being a relatively new technology, it has garnered huge attention from the research community, leading to the development of several open-source simulators [6], [15], [16]. The simulators support essential features of FL, such as heterogeneity, resource limitations, non-Independent and Identically Distributed (IID) data, and communication constraints. However, traditional FL simulators do not model the intricacies of a VANET scenario. Modeling the interactions of VANET participants is essential to derive reliable insights about the communication load in ITS. Hence, traditional FL simulators are not sufficient to capture the intricacies of FL in ITS applications.

A simulator to study FL-based ITS applications requires modeling of mobility, network, and FL behavior. The mobility and network components combination is readily available as a VANET simulator. Hence, significant development efforts can be saved by extending an existing VANET simulator with support for FL. An example of this approach is the implementation of FLEXE [8] as an extension of Veins [9]. FLEXE is demonstrated with a training example of Multilayer Perceptron (MLP) model using 200 vehicles.

The computational complexity limits the scenario's scale because of the high-fidelity models of Veins. As the focus is on the FL behavior, though, simplifying network or mobility components is a reasonable choice. This reduces the computational complexity and allows high-fidelity modeling of FL behavior. A network simulator can act as a VANET simulator with mobility simplified to vehicular traces. Authors of [17] developed a coupled simulation framework using ns-3 and FLSim [7]. ns-3 is a high-fidelity network simulator that supports 5G through an extension [18]. However, ns-3 is computationally complex because it is geared toward protocol research. Indeed, the experiments confirm that ns-3 is the bottleneck [17].

Alternatively, a mobility simulator can be coupled with an FL simulator. An example is the city-scale study of parking lot occupancy prediction using FL [19]. Authors use SUMO [20] as the mobility simulator. Interestingly, the demonstration scenario is on a city scale. However, the model training happens once per simulation day, which implies a substantially lower training load. This allowed the authors to focus only on speeding up the SUMO simulations by employing extra computing resources. A higher model training load increases the computation requirements and prevents the ability to study city-scale scenarios.

A high-fidelity network or mobility simulator is a good choice for small-scale scenarios [8], [17] or scenarios with infrequent model training [19]. However, this does not match some of the use cases envisioned for FL in VANET, wherein it is expected to train multiple models several times throughout the day [21]. The advanced applications of ITS, such as Autonomous Vehicles (AVs), require regular model updates to tackle the evolving traffic and weather conditions. Hence, simplifying only one of the components of VANET fails to improve the scalability.

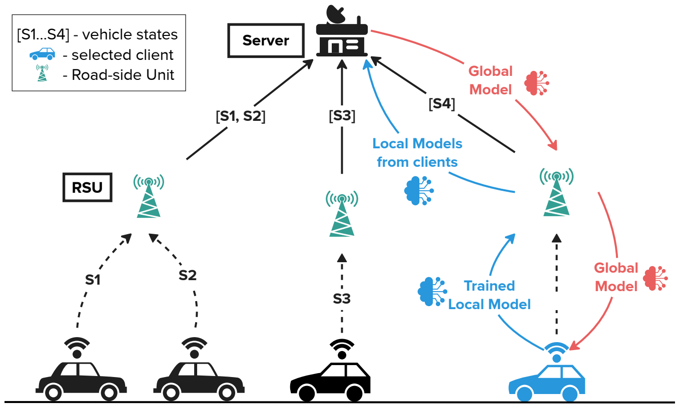


Figure 1. Federated Learning in a VANET scenario

Simplifying mobility and network components allows high-fidelity modeling of FL. This approach is common in problem-specific simulators that are increasingly being proposed to evaluate 5G technologies for ITS. The network slicing research community employs a simulator specifically for VANET slice management [22]. V2XArcSim enables infrastructure management studies for VANETs [23]. In addition to technology-specific simulators, application-specific simulators have also been designed. For example, a special simulator is proposed to study autonomous driving with 5G [24]. The targeted approach of designing simulators compromises on generalization but offers scalability. As a result, the simulator can focus on modeling the problem under study in high fidelity and conduct large-scale evaluations.

FL, though, is a technology that is expected to play a key role in ITS with several open research questions [21]. This warrants a dedicated simulator to study FL in the context of ITS. This paper addresses the gap by proposing an integrated simulation approach to study ITS applications based on FL. We propose using the VANET simulator with the simplified network and mobility components to support large-scale evaluations. We validate our implementation and demonstrate that the traditional FL simulator falls short in modeling the constraints of VANET environment.

III. Federated Learning in VANETs

In a traditional Machine Learning (ML) model training, raw data from various sources is collected at a central location. After pre-processing the data, the model is trained and utilized for inference. However, privacy and communication constraints prevent access to raw data from all available clients. To mitigate these concerns, an alternative strategy is proposed in the form of FL [25]. In FL, a central server is designated to train the ML model using a federation of clients.

In the VANET context, vehicles act as clients, and a central traffic controller acts as a server. FL in a typical urban VANET scenario is shown in Figure 1. Roadside Units (RSUs) are distributed across the road network and are designed to have a stable connection to the server. Vehicles interact with the RSUs and share their state information, which is then forwarded to the server. The server selects a set of vehicles for training using the state information. The local model is transmitted to the relevant RSUs so they can forward it to the selected vehicles. Once the model training is completed on the vehicle, the updated local model is transmitted to any RSU that the vehicle can communicate with. Note that it need not be transmitted to the same RSU that initially sent the global model to begin training. Once the training round is completed, local models are forwarded to the RSUs, which are further forwarded to the server.

Modeling FL needs several considerations, such as client selection, model aggregation, data heterogeneity, system heterogeneity, communication constraints, etc. These are discussed in detail by the traditional FL simulators and their respective documentation [6], [7]. Since we are modeling FL for ITS applications, it is essential to adapt these considerations for the VANET environment.

A. State

Traditional FL simulators often create clients on the fly and destroy them once the training round is completed. Such an implementation does not meet the requirements of VANET agents. Typically, VANET behavior is modeled such that the agent has a persistent state for the entire duration of its trip. Moreover, the FL training in a VANET environment has to be opportunistic by utilizing the available clients. It is not reasonable to create vehicles on the fly to have sufficient participants in the training. Hence, the behavior of the client and server as a VANET agent must not depend on its role in the FL training. To achieve this, our approach models the agents as first and foremost VANET agents with the FL behavior as an extension. The server initiates the activities of the FL, and the clients must react and update their state accordingly.

B. Discrete-time paradigm

Traditional FL simulators do not consider discrete-time paradigms. The execution time is determined by the number of training rounds [6], [7]. On the other hand, VANET simulation is typically modeled using a discrete-time paradigm with agent-based model (ABM) [9]. Since our approach relies on extending VANET agents, a discrete-time paradigm with ABM is incorporated. Another benefit is this implementation allows expansion of the FL behavior to consider the practical aspects of VANET. For instance, all the individual steps of FL, client analysis, client selection, training rounds, and model aggregation can be designed to take a configurable amount of time. As a result, the training failures due to client dropouts are naturally induced due to the VANET behavior instead of random probabilistic choices.

C. Vehicle Sensors

Traditional FL simulators do not consider the sensor behavior of the vehicles in data allocation. At the beginning of the simulator, the datasets are divided into partitions, which are distributed to the clients. Later, when a client is selected for training, the entire partition is immediately available for use. However, this is not the only mode in which VANET clients behave. Hence, the simulator should support an additional behavior where the samples are available gradually over time. The two possible configurations in which VANET clients can be modeled are:

- *Immediate*: The entire partition is immediately available for training. This is similar to the default behavior of traditional FL simulators. This approach is valid for scenarios where the model is infrequently trained, or sensors are event-based with sparse data collection. In such scenarios, it can be assumed that the vehicles collected the necessary data on their previous trip.
- *Sensor-like*: The client should gradually mark a configurable fraction of samples from the partition as available for training. This matches the vehicle behavior because vehicles collect the sensor data gradually as they drive. The approach is suitable when the sensor data is collected at regular and high-frequency intervals. However, the limited onboard storage can prevent the vehicles from storing the data longer [26]. Hence, the collected samples must be allowed to go stale after a configured period.

D. Heterogeneity

The decentralized nature of FL gives rise to heterogeneity. There are four types of heterogeneity: statistical, model, communication, and device [27]. VANET scenarios express all forms of heterogeneity, but it depends on the VANET behavior. For instance, Gaussian noise or blur is added as feature skew in FL to account for sensor variations among the clients [28]. In VANET, this can be modeled as an agent behavior instead of a random process. The amount of noise and blur added to the features can be modeled based on the vehicle's properties, like speed, manufacturer, or sensor.

E. Client Selection

At each training round, the server must select a subset of clients for the training process. Common proposals of FL suggested utilizing a randomly sampled subset of available clients [25]. Random selection works for most cases. However, the training process can be improved by exploiting certain client characteristics. Several client selection processes are proposed to exploit the system and data heterogeneity [29]. Traditional FL simulators have default implementations for many popular client selection techniques. Our approach supports adapting these client selection methods for VANET scenarios.

In addition to the heterogeneity aspects considered for client selection, agent-like vehicle behavior should be supported. This might be as simple as selecting only vehicles of a particular manufacturer for training or as complex as ensuring that a vehicle approaching its destination might not be selected for training as it may exit the road network before uploading the local model. Many similar approaches might be considered worthwhile to investigate: A vehicle that has just entered the network might not be preferred because very little sensor data is available for training. Emergency vehicles may or may not be included in the selection. Vehicles in one sub-region may be preferred over another because of the interesting dynamic conditions like weather or traffic. During the non-driving period, the vehicles might be inactive and cannot be selected for training. Our proposed simulator supports the configuration of client selection methods to respect these vehicular aspects.

F. Communication

Communication is modeled as a random process in a traditional FL simulator. However, this fails to capture the actual VANET behavior. Moreover, the control over the communication aspects of the simulation is limited. Our proposed simulator stems from the VANET simulator as an extension. As a result, the behavior of the VANET can be accurately captured. Further, this allows reliable estimation of the impact of FL data loads on the entire ITS infrastructure. Because of the ABM implementation, what-if questions can be explored in the VANET parameter space. For instance, *what* is the impact on training performance *if* 25% of RSUs are offline?

IV. IMPLEMENTATION

A. VANET Simulator

We implement the proposed simulator with some of the primary features described in Section III. The required modification is that the VANET agent behavior must be extended to perform the FL client and server tasks. There are several potential VANET simulators that can be selected for implementing our approach [9], [30], [31]. A simplified VANET simulator is preferred for its scalability. It must be noted that the simplified simulator results in insights indicative of a steady-state network behavior. Despite this limitation, the advantage is that it enables a holistic steady-state evaluation of the system-wide impact. For a complex evaluation that focuses on resilience, a high-fidelity simulator like FLEXE can be employed [8].

Another benefit of using a simplified but scalable VANET simulator is that this leaves room to model FL behavior in high fidelity. The use cases for ITS applications are expected to train AI models several times throughout the day [21]. Further, multiple AI models need training because of the various applications to serve. Hence, simplifying the VANET simulations enables studying scenarios with a high model training load.

In our approach, a simplified VANET simulator called *Disolv* is used [31] which allows ITS investigations such as optimization studies [32] at scale. *Disolv* returns comparable results in steady state compared to other high-fidelity VANET simulators. The agent model of *Disolv* is extended to support FL client and server models. This results in an integrated simulator implementation.

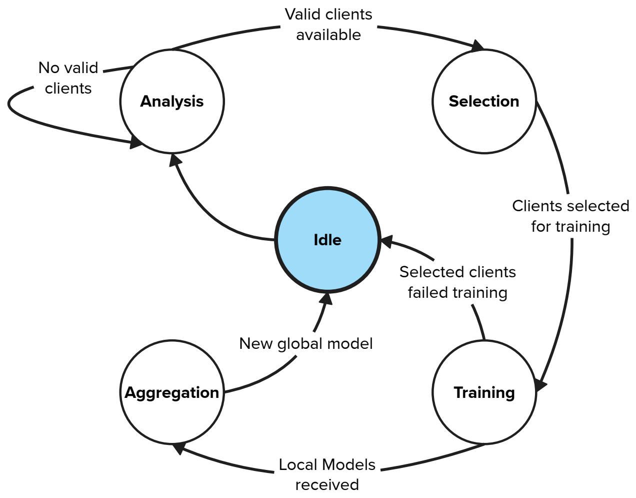


Figure 2. State diagram of a Federated Learning Server

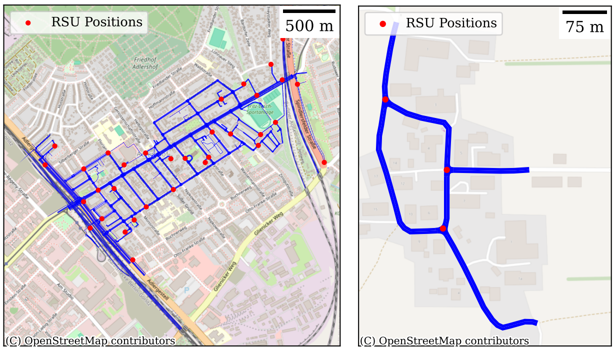
B. Server

The state diagram for a server is shown in Figure 2. Every server is initialized with the *Idle* state, and it performs VANET activities until it is time to begin the FL procedure. The server collects the state information about clients in the *Analysis* state. If there are valid clients, the server changes to the *selection* state. Otherwise, it continues in the *Analysis* state. At least one client is selected for training in each round, and the selected clients are informed to initiate the training process. Once the pre-defined training duration is completed, the server moves to the *Aggregation* state and collects the local models. If all the selected clients fail the training process, the *Aggregation* step is skipped. After aggregating the models, the server goes back to the *Idle* state.

The state transition condition depends on another factor, which is the time spent in each state. The user can define a specific time to spend in each state. As a result, the state transitions are spaced out over a period. The *Sensor-like* behavior mimics the VANET agents collecting the sensor data gradually. By spreading out the server tasks, vehicles are allowed sufficient time to collect sensor data for training. Another benefit is that the clients and server can break their model uploads into chunks and complete them within the designated time of the state. This also mimics the real world, where the server performs various tasks. The desired model training can be assumed to be one of the tasks the server carries out based on a schedule.

C. Client

Clients are designed to be simple and merely respond to the server's commands. When a client is selected for training, the server sends the global model as the starting point. Using the global model, the client begins the training process and builds an updated local model. Once the server requests the training



(a) *district* scenario (b) *village* scenario

Figure 3. Road networks used in the simulations along with the RSU positions (Map data from OpenStreetMap, openstreetmap.org/copyright).

to conclude, the client uploads the trained model. Each client carries out these activities along with its VANET behavior.

D. Flexibility

The VANET agent implementation is extended to support FL behavior. Every agent, irrespective of their role in the VANET simulation, can be assigned the role of either FL client or server. This allows flexibility in modeling a variety of FL training scenarios. An RSU can act as an FL server and control a training process with vehicles in its vicinity as clients. Similarly, a central server can act as an FL server and train the models on RSUs. A cooperative training scenario can be designed among the vehicles, with one of the vehicles acting as a server. Combining different approaches into a multi-model training scenario is also possible.

V. Experiments

A. Scenarios

A simple federated training of the MNIST model is selected to demonstrate the capabilities of our proposed simulator. Initial experiments validate the simulator, and further experiments demonstrate the impact of VANET behavior in FL. By default, the dataset is distributed uniformly across the clients, mimicking an IID scenario. For the VANET aspects, a typical VANET environment is created. The road network consists of RSUs distributed over the network, most placed at the intersections. RSUs are assumed to have a fixed connection to the central server. Vehicles communicate periodically as they traverse the road network. Each vehicle is allocated with a single trip over the simulation. Two different scenarios with varying sizes of road networks called *village* and *district* scenarios are designed.

1) village *scenario:* This is a simple network with 3 RSUs and 100 vehicles. The simulation duration is set to 600s, and the time step is 0.1s.

2) district *scenario:* This is a slightly larger road network with 33 RSUs and 2218 vehicles. The simulation duration is set to 3800s, and the time step is 0.1s.

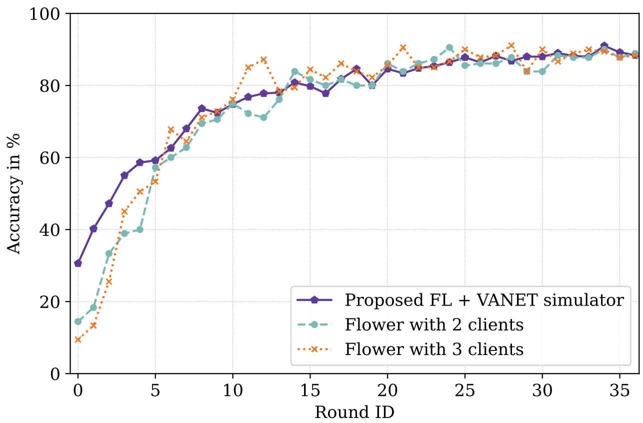


Figure 4. Validation of mean global model accuracy obtained from 10 runs for *district* scenario. Note that the number of clients and rounds in Flower are configured based on the results from the proposed simulator.

Table I
Timing Configuration for FL Server

Parameter	Time in Seconds	
	village scenario	*district* scenario
Initiation duration	20	63
Analysis duration	2	2
Selection duration	2	2
Training duration	15	30
Aggregation duration	5	2

The road networks are shown in Figure 3. The timing settings for the FL server are mentioned in Table I. The default model settings are mentioned in Table II. The default client selection method is random, and the default aggregation method is *FedAvg* [25]. All the experiments use these settings unless mentioned otherwise. In the following subsections, we describe the experiments and their important takeaways.

B. Validation

This experiment aims to build an FL scenario in the proposed simulator and replicate the same in a traditional FL simulator for validation purposes. Flower is selected as the traditional FL simulator for comparison. The dataset allocation is set to *Immediate* because it is the default in Flower. We select the MNIST model for training in the *district* scenario. Because of our simulator's emergent behavior, pre-determining the FL parameters, such as the number of clients and rounds, is impossible. Hence, we initially run the simulations in our implementation and observe these parameters. Later, the Flower experiments are configured accordingly.

In this scenario, the selected clients varied between 2 and 3, with 3 being the most common. This variation in trained clients is possible because the vehicles begin and end their trips throughout the simulation. Further, the number of possible training rounds is found to be 37. This is due to selecting the FL server timings as per Table I. Hence, we performed the

Table II
PARAMETER SETTINGS FOR MNIST MODEL TRAINING

Parameter	Value
Number of epochs	3
Batch size	100
Learning rate	0.0001
Number of Classes	10
Dropout	0.5
Random share in *village*	0.2
Random share in *district*	0.04

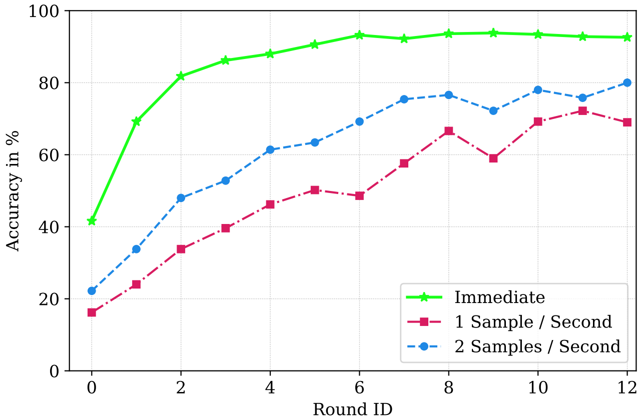


Figure 5. Effect of sensor data allocation behavior on model accuracy

Flower experiment for 37 rounds with 2 client and 3 client configurations.

The validation results are shown in Figure 4. The global model accuracy is the mean value of 10 experiment runs. Despite the initial variations, the accuracy of all three simulations eventually converges to a very close value. In the following subsections, we investigate the effects of VANET behavior on the FL training.

C. Sensor Data Behavior

The goal of this experiment is to determine the impact of the *Sensor-like* behavior of vehicles in collecting the samples. MNIST model is trained in the *village* scenario. The dataset allocation is set to *Sensor-like* with two configurations. In the first configuration, vehicles collect 1 sample per second. In the second configuration, vehicles collect 2 samples per second. As a baseline, the *Immediate* setting is used to mimic the traditional FL simulator behavior.

The experiments are carried out to determine mean global accuracy over 10 experiment runs per setting. The results are shown in Figure 5. The *Sensor-like* configuration allows the gradual collection of data, resulting in fewer samples for training. The server timing configuration in Table I also plays a role in this behavior. The data partitions are larger in the *village* scenario with about 600 samples per vehicle. As a result, the vehicles have little time to collect their entire partition.

On the other hand, the *district* scenario has many vehicles, resulting in smaller data partitions per vehicle (26). Hence, for the selected initiation time, vehicles have enough time to collect most of the samples from their partitions. As a result, the impact of *Sensor-like* behavior is minimal in the *district* scenario. The global accuracy is comparable for all the experiments, and the figure is omitted for brevity. This experiment clearly shows that the VANET agent behavior affects FL training.

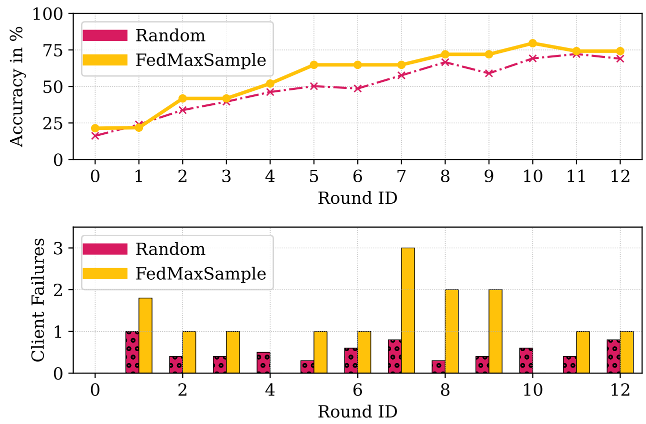


Figure 6. Impact of VANET behavior on client selection methods

D. Client Selection

Heterogeneity is typical of an FL scenario. The default client selection can be replaced with a more intelligent one to improve the convergence speed. Many methods exist to optimize the selection process based on various clients and their data characteristics [33]. The simplest method is based on the statistical utility of the client, which suggests selecting clients with the maximum sample count. We define this method as *FedMaxSamples*. We compare the training performance with random client selection and *FedMaxSamples*. An MNIST model is trained in the *village* scenario. The partition method is set to *Sensor-like* with 1 sample per second configuration.

The average global model accuracy over the 10 runs is shown in Figure 6. Along with the accuracy, the average client failures are compared on the second y-axis. As expected, the *FedMaxSamples* method improves the model accuracy but introduces a side effect. Client failures are the number of selected clients that failed to upload the local model to the FL server. Because of the variations between the rounds, the client failure average taken over multiple runs will not be a whole number. With random selection, there were 6.5 failures on average. Whereas, with *FedMaxSamples* selection, the failures increased to 14.8. Vehicles close to completing their trip have maximum samples, and *FedMaxSamples* selects these clients for training. Hence, the vehicles have little time to complete the training, resulting in client failures.

A combination of VANET agent behavior, the server timings, and the *FedMaxSamples* design are responsible for this behavior. The results demonstrate that our proposed simulator can capture these inter-dependent phenomena of FL and VANET.

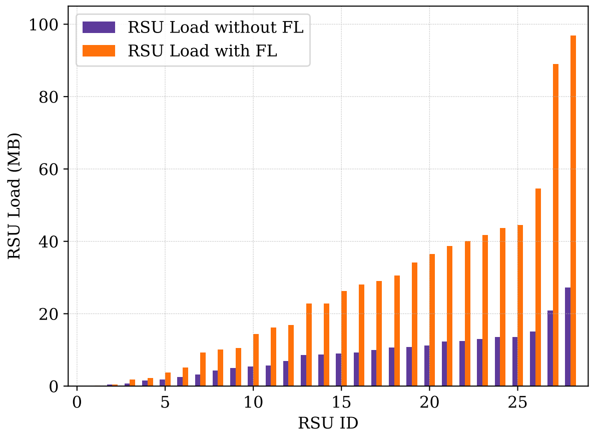


Figure 7. Comparison of RSU transmission load with and without FL in the *district* scenario.

Table III
EXECUTION TIME COMPARISON FOR THE "DISTRICT" SCENARIO

Scenario	Execution time
FL scenario (Flower)	31 s
FL+VANET scenario (our approach)	120 s

E. Communication Results

Capturing the impact of VANET is another crucial goal for the proposed simulation approach. The hardware deployed for ITS is managed by 5G and beyond protocols. VANET is expected to have its own network slice [22]. This experiment aims to understand the additional data load that the ITS infrastructure has to support because of the FL introduction. The connection from RSUs to FL server is assumed to be stable. Hence, the interesting part is the wireless communication between RSU and the vehicles. We compute the total transmission load of the RSUs. The insights will allow estimating changes for the VANET network slice to avoid disrupting the FL training.

MNIST model is trained for *district* scenario. An experiment with no FL training is simulated as a baseline. The vehicles and RSU exchange default information of size 100 bytes every time step. In the FL scenario, in addition to the default information, the FL related data like the AI models are also transmitted. The RSU load comparison is shown in Figure 7. The load increase is marginal in all but two RSUs. The total data transfer increase is ~526MB for the entire simulation period of 3800 seconds. Accordingly, the placement of the RSUs can be adjusted to alleviate the load on these two specific RSUs. This experiment highlights the capabilities of the proposed simulator in capturing the impact on the infrastructure due to FL training.

F. Runtime Performance

Several techniques are proposed to accommodate ITS environment constraints in FL training [34], [35]. However, the evaluations are rarely carried out on a large scale. One of the reasons is the higher performance cost of simulating both FL and VANET together [8]. Our implementation is based on a simplified VANET simulator with a relatively low computational cost. Table III shows execution times for the *district* scenario with Flower and with our approach. Compared to 31 seconds for Flower (which does not consider VANET behavior), with 2200 agents, our approach takes 120 seconds. Hence, for a reasonable increase in execution time, our simulator can study the combined effects of FL and VANET. As a result, complex algorithms for FL techniques can be studied at scale. Further, there is more room for training load to consider larger ML models.

VI. Discussion

Traditional FL simulators are built to model any external factors as random distributions. VANET behavior is intricate and requires ABM to capture the complex interactions. The emergent phenomena of VANETs also impact the outcomes in the FL training. Such complex interactions demand a different perspective to evaluate. Our proposed approach has several benefits, with the main contribution being the ability to study the combined effects of FL and VANET at scale.

In addition to the realistic FL modeling, there are benefits from the VANET perspective. Our simulator can be used to understand the communication side of the FL in greater detail. The network capacity can be evaluated to determine if it is sufficient for the FL scenarios. Further, it is possible to pinpoint the exact region and the entity that needs attention. As shown in Figure 7, two RSUs see a comparatively higher data load. Additional RSUs can be deployed to share their data load and avoid communication failures. In addition to the network constraints, new modes of operation can be evaluated for feasibility. Collaborative FL model training scenarios among the participants of the VANET can be studied. This can include exploiting the resources of parked cars as edge computing devices and offloading the training process to them. Further, a decentralized FL server can be considered because of the geographical constraints in large regions.

One of the main limitations of our simulator is that, by design, the results are indicative only of a stable state of the system. Network overload effects, such as latency and packet failures, cannot be modeled using a simplified simulator. This makes it unsuitable to investigate use cases that focus on, e.g., communication technology research, instead of larger-scale network behavior.

VII. Conclusion

This paper proposes a new integrated simulation approach for investigating Federated Learning (FL) in Vehicular Ad-hoc Networks (VANETs) at scale. The proposed approach extends our large-scale Intelligent Transportation System (ITS) simulator *Disolv* to also model FL behavior. The validation experiment with Flower shows that (in cases with simplified VANET behavior that both simulators can model) both simulators produce comparable results. Further, the sensor behavior and client selection comparisons demonstrate the impact of true VANET behavior on FL training. An investigation of runtime performance shows that the approach incurs only a reasonable increase in execution time compared to much more simplified studies. To foster both FL and larger-scale ITS research, we share our implementation of the proposed approach as open source.[1]

References

[1] J. Lee, F. Solat, T. Y. Kim, and H. V. Poor, "Federated Learning-Empowered Mobile Network Management for 5G and Beyond Networks: From Access to Core," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 2176–2212, 2024.

[2] Y. Lim, J. Edelenbos, and A. Gianoli, "What is the impact of smart city development? Empirical evidence from a Smart City Impact Index," *Elsevier Urban Governance*, vol. 4, no. 1, pp. 47–55, Mar. 2024.

[3] R. Zhang, J. Mao, H. Wang, B. Li, X. Cheng, and L. Yang, "A Survey on Federated Learning in Intelligent Transportation Systems," *IEEE Transactions on Intelligent Vehicles*, pp. 1–17, 2024.

[4] M. Blumreiter, J. Greenyer, F. J. Chiyah Garcia, V. Klös, M. Schwammberger, C. Sommer, A. Vogelsang, and A. Wortmann, "Towards Self-Explainable Cyber-Physical Systems," in *22nd IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MODELS 2019), 14th International Workshop on Models@run.time (MRT 2019)*, Munich, Germany: IEEE, Sep. 2019.

[5] S. Sharma, R. Popli, S. Singh, G. Chhabra, G. S. Saini, M. Singh, A. Sandhu, A. Sharma, and R. Kumar, "The Role of 6G Technologies in Advancing Smart City Applications: Opportunities and Challenges," *MDPI Sustainability*, vol. 16, no. 16, p. 7039, Aug. 2024.

[6] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, "Flower: A Friendly Federated Learning Research Framework," *arXiv preprint arXiv:2007.14390v5*, 2020.

[7] L. Li, J. Wang, and C. Xu, "FLSim: An Extensible and Reusable Simulation Framework for Federated Learning," in *Simulation Tools and Techniques*, Springer, 2021, pp. 350–369.

[8] W. Lobato, J. B. D. da Costa, A. M. de Souza, D. Rosário, C. Sommer, and L. A. Villas, "FLEXE: Investigating Federated Learning in Connected Autonomous Vehicle Simulations," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, IEEE, Sep. 2022, pp. 1–5.

[9] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[10] X. Kong, L. Lu, and K. Xiong, "Privacy-Preserving Estimation of Electric Vehicle Charging Behavior: A Federated Learning Approach Based on Differential Privacy," *Elsevier Internet of Things*, vol. 28, p. 101 344, Dec. 2024.

[11] X. Zhou, Z. Liao, Y. Zhao, Y. Liu, and A. Yi, "Ride-Hailing Pick-up Area Recommendation in a Vehicle-Cloud Collaborative Environment: A Feature-Aware Personalized Clustering Federated Learning Approach," *Springer Cluster Computing*, vol. 28, no. 1, Oct. 2024.

[12] Y. Li, X. Xu, G. Huang, M. Yao, L. Sun, and J. Xu, "VSFL: Trajectory Prediction Framework Based on Validity-Aware Semi-Asynchronous Federated Learning in Internet of Vehicles," *Elsevier Computer Communications*, vol. 224, pp. 106–117, Aug. 2024.

[13] Z. Yang, C. Cheng, Z. Li, R. Wang, and X. Zhang, "Reliable federated learning based on delayed gradient aggregation for intelligent connected vehicles," *Elsevier Engineering Applications of Artificial Intelligence*, vol. 140, p. 109 719, Jan. 2025.

[14] Y. Cui, J. Zhu, and J. Li, "FLAV: Federated Learning for Autonomous Vehicle Privacy Protection," *Elsevier Ad Hoc Networks*, vol. 166, p. 103 685, Jan. 2025.

[15] D. Zeng, S. Liang, X. Hu, H. Wang, and Z. Xu, "Fedlab: A flexible federated learning framework," *Microtome Publishing Journal of Machine Learning Research*, vol. 24, no. 100, pp. 1–7, 2023.

[16] F. Lai, Y. Dai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "FedScale: Benchmarking Model and System Performance of Federated Learning," in *Proceedings of the First Workshop on Systems Challenges in Reliable and Secure Federated Learning*, ser. SOSP '21, ACM, Oct. 2021.

[17] E. Ekaireb, X. Yu, K. Ergun, Q. Zhao, K. Lee, M. Huzaifa, and T. Rosing, "ns3-fl: Simulating Federated Learning with ns-3," in *Proceedings of the 2022 Workshop on ns-3*, ACM, Jun. 2022, pp. 97–104.

[18] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G – An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, Jan. 2020.

[19] L. Alekszejenkó and T. Dobrowiecki, "SUMO Simulations for Federated Learning in Communicating Autonomous Vehicles: A Survey on Efficiency and Security," in *SUMO Conference Proceedings*, vol. 4, TIB Open Publishing, Jun. 2023, pp. 115–129.

[20] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic Traffic Simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Nov. 2018, pp. 2575–2582.

[21] V. P. Chellapandi, L. Yuan, S. H. Żak, and Z. Wang, "A Survey of Federated Learning for Connected and Automated Vehicles," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Sep. 2023, pp. 2485–2492.

[22] C. Zamfirescu, R. Iugulescu, R. Crăciunescu, A. Vulpe, F. Y. Li, and S. Halunga, "Network Slice Allocation for 5G V2X Networks: A Case Study from Framework to Implementation and Performance Assessment," *Elsevier Vehicular Communications*, vol. 45, p. 100 691, Feb. 2024.

[23] R. Chintalapati, N. D. Tripathi, and J. H. Reed, "V2XArcSim: Evaluation of Efficient and Scalable 5G-based V2X Infrastructure Architectures," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, IEEE, Oct. 2024.

[24] M. Jooriah, D. Datsenko, J. Almeida, A. Sousa, J. Silva, and J. Ferreira, "A Co-Simulation Platform for V2X-Based Cooperative Driving Automation Systems," in *2024 IEEE Vehicular Networking Conference (VNC)*, IEEE, May 2024, pp. 227–230.

[25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[26] A. Khalil, M. Lotfian Delouee, V. Degeler, T. Mcuser, A. F. Anta, and B. Koldehofe, "Driving Towards Efficiency: Adaptive Resource-Aware Clustered Federated Learning in Vehicular Networks," in *2024 22nd Mediterranean Communication and Computer Networking Conference (MedComNet)*, IEEE, Jun. 2024, pp. 1–10.

[27] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous Federated Learning: State-of-the-art and Research Challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, Oct. 2023.

[28] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated Learning on Non-IID Data Silos: An Experimental Study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, IEEE, May 2022, pp. 965–978.

[29] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client Selection in Federated Learning: Principles, Challenges, and Opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 811–21 819, Dec. 2023.

[30] B. Schünemann, "V2X Simulation Runtime Infrastructure VSimRTI: An Assessment Tool to Design Smart Traffic Management Systems," *Elsevier Computer Networks*, vol. 55, no. 14, pp. 3189–3198, Oct. 2011.

[31] N. T. Tangirala, C. Sommer, and A. Knoll, "Simulating Data Flows of Very Large Scale Intelligent Transportation Systems," in *Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS 2024)*, ACM, Jun. 2024, pp. 98–107.

[32] N. T. Tangirala, C. Sommer, and A. Knoll, "Optimizing Very Large Scale ITS Applications With Fast Fitness Evaluation," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2025, pp. 1–6.

[33] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client Selection in Federated Learning: Principles, Challenges, and Opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 811–21 819, Dec. 2023.

[34] Y. Ren, T. Wang, S. Zhang, and J. Zhang, "An Intelligent Big Data Collection Technology Based on Micro Mobile Data Centers for Crowdsensing Vehicular Sensor Network," *Springer Personal and Ubiquitous Computing*, vol. 27, no. 3, pp. 563–579, 2023.

[35] Z. Yang, X. Zhang, D. Wu, R. Wang, P. Zhang, and Y. Wu, "Efficient asynchronous federated learning research in the internet of vehicles," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7737–7748, 2022.

[1] Source code at: https://github.com/nagacharan-tangirala/disolv