

# Joint Air-Ground Distributed Federated Learning for Intelligent Transportation Systems

Swapnil Sadashiv Shinde<sup>1b</sup>, *Student Member, IEEE*, and Daniele Tarchi<sup>2b</sup>, *Senior Member, IEEE*

**Abstract**—Supported by some of the major revolutionary technologies, such as Internet of Vehicles (IoVs), Edge Computing, and Machine Learning (ML), the traditional Vehicular Networks (VNs) are changing drastically and converging rapidly into one of the most complex, highly intelligent, and advanced networking systems, mostly known as Intelligent Transportation System (ITS). Recently, distributed ML techniques, such as Federated Learning (FL) have gained huge popularity mainly for their advantages in terms of intelligence sharing and privacy concerns. VNs are a natural contender for exploiting FL for solving challenging problems; however, their limited resources, dynamic nature, high speed, and reduced latency requirements often become the bottleneck. V2X communication technologies allow vehicular terminals (VTs) to share their valuable local environment parameters and become aware of their surroundings. Such information can be utilized to build a more sustainable and affordable FL platform for serving VTs. Gaining from recently introduced 3D architectures, integrating terrestrial and aerial edge computing layers, we present here a distributed FL platform able to distribute the FL process on a 3D fashion while reducing the overall communication cost for providing vehicular services. The framework is defined as a constrained optimization problem for reducing the overall FL process cost through a proper network selection between various nodes. We have modeled the FL network selection problem as a sequential decision-making process through a Markov Decision Process (MDP) with time-dependent state transition probabilities. A computation-efficient value iteration algorithm is adapted for solving the MDP. Comparison with various benchmark methods shows the overall improvement in terms of latency, energy, and FL performance.

**Index Terms**—Vehicular edge computing, federated learning, aerial networks, Markov decision process.

## I. INTRODUCTION

**I**N THE last decade, Multiaccess Edge Computing (MEC), being one of the most recent revolutionary technology, has enabled various latency limited and data-intensive services and applications in the wireless networking scenarios [1]. In the case of Vehicular Networks (VNs), edge computing can be

enabled through the implementation of Roadside Units (RSUs) and integrating them with limited capacity edge servers, known as Vehicular Edge Computing (VEC) [2]. However, with growing service requirements, limited RSUs resources are not sufficient and are becoming a bottleneck for VEC performance. Recently, different aerial platforms, such as Low and High Altitude Platforms (LAPs and HAPs), Unmanned Aerial Vehicles (UAVs), drones, and balloons, are integrated into the terrestrial networks creating a single Terrestrial/Non-terrestrial (T/NT) network [3]. These platforms can be further exploited for their reliable computation and communication capabilities through the edge computing paradigm. If integrated into the VNs, the aerial platforms can boost the performance of traditional VEC facilities with additional resources.

In recent times, the Internet of Vehicles (IoV) paradigm has been introduced, where VNs, through the integration of communication and sensing, become a large source of data [4]. These data can be analyzed through different Machine Learning (ML) techniques for providing higher Quality of Service (QoS) and Quality of Experience (QoE) in vehicular services with reduced costs to the end-users. ML algorithms can find the hidden patterns and underlying structures in data collected by Vehicular Terminals (VTs) without human intervention. Therefore, ML techniques, including Deep Learning (DL), Reinforcement Learning (RL), and Federated Learning (FL), have been suggested to be used to solve challenging research problems on VNs in the recent years [5], [6], [7]. Compared with the traditional centralized ML process, FL had enormous success in terms of reduced latency and energy performance [8], [9]. In general, a FL process involves several training iterations, also known as FL iterations, characterizing a training process over distributed FL-clients (i.e., wireless nodes), transmission of updated local ML model parameters towards a centralized server, an averaging operation to be performed over a centralized FL-server for creating a global model (i.e., Federated averaging (FedAvg)), and the transmission of new global model parameters towards FL-clients [10]. Through local training operations, wireless devices can save energy and reduce data transmission delay when sending raw data towards a centralized server. Additionally, through an aggregation process performed at a centralized FL-server, devices can learn from each other's training experience. Though FL has these advantages, implementing in distributed environments, such as VNs, can be challenging mainly due to, e.g., the resource scarcity of individual VTs, involvement of high mobility, dynamically changing vehicular environments.

Manuscript received 29 September 2022; revised 31 January 2023 and 16 March 2023; accepted 4 April 2023. Date of publication 12 April 2023; date of current version 30 August 2023. This work was supported in part by the European Union through the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future," under Grant PE00000001-program "RESTART." The Associate Editor for this article was S. Wan. (*Corresponding author: Daniele Tarchi.*)

The authors are with the Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi," University of Bologna, 40136 Bologna, Italy (e-mail: swapnil.shinde2@unibo.it; daniele.tarchi@unibo.it).

Digital Object Identifier 10.1109/TITS.2023.3265416

The main challenges, when implementing the FL platform over a distributed VNs, are:

- The harsh wireless environment due to vehicular mobility can make it hard to implement the traditional centralized FL model in VNs.
- VTs' resource limitations can be a bottleneck for performing a large number of FL iterations.
- Uncertainty about the participation of a large number of VTs into the FL process, mainly due to the harsh wireless environment, can also increase the FL process cost.
- Continuously changing vehicular environment parameters and corresponding data can impact the FL model performance over time, and introduce model drift. This may require frequent evaluation and retraining of the FL model.

Though FL has enormous potentials, the vehicular community is hesitant to embrace FL, mainly for these challenges. Some recent works have however considered the applications of FL over VNs while doing some assumptions [11]. A particular attention has been focused on multi-layered edge computing enabled VNs, capable of communicating with RSUs, UAVs, and HAPs, able to tackle the long-distance communication challenge introduced by the centralized FL models by distributing the learning process over multiple nodes (i.e., distributed FL) [12]. With limited range communications, VTs can also save energy/resources during the FL iterations, allowing to perform more iterations [13]. Failures of communication links between VTs and FL servers can be reduced, and a large number of VTs can participate in the FL process [14]. Further, taking into account the VTs' local environment and the application requirements, through various scenario-based modelings, the implementation of the FL process can become more reliable and sustainable over a dynamic vehicular environment [15].

In this work, we propose a distributed FL process for vehicular applications. We consider a joint T/NT network composed of VTs, RSUs, UAVs/LAPs, and HAPs. A distributed FL platform allowing a distributed and flexible FedAvg process is proposed exploiting different layers, where RSUs, UAVs, and HAPs are capable of collecting the FL model updates from VTs in each FL iteration for generating the global update vectors/models for the next iterations.

With the possibility of exploiting multiple layers, a proper selection of the nodes where the FL process should be performed is required, aiming at enhancing latency, energy, and FL process performance. However, solving such a network selection problem over a multi-layered dynamic VN can be enormously complex, and traditional optimization techniques are inefficient. In the recent past, various works have highlighted the importance of RL-based approaches for solving the network selection problem over VNs [16]. In this work, we resort to the RL method for solving the FL network selection problem. As a first step, we model the problem as a sequential decision-making process through a Markov Decision Process (MDP) framework which requires a proper design of a state space, action space, reward function, and environment dynamics [17]. The environment dynamics are modeled through a set of time-dependent state

transition probability expressions, considering the VTs local environment for designing the MDP with better performance.

The main contributions of this work can be summarized in the following points:

- We design a communication efficient distributed FL model over a joint air-ground network aiming at reducing FL process latency and energy costs.
- A constrained optimization problem is formulated for minimizing the overall cost (in terms of joint latency, energy and the FL training performance) of the process by a proper assignment of VTs and FL servers.
- A MDP framework is considered for modeling the problem as a sequential decision-making process, and a value iteration technique is used to solve it. The MDP environment is modeled through time-dependent state transition probabilities that take into account the local vehicular environment.
- The performance of the proposed scheme is analyzed by comparing it with different heuristic techniques and conclusions are drawn.

## II. RELATED WORKS

Traditional VNs are converging into a more intelligent and advanced networking system with the integration of edge computing, machine learning, and big data applications [2]. Though VEC brings several advantages by enabling innovative services in the VN, the limited storage and computation capacity of edge servers is becoming a bottleneck for the innovative ML-based applications. The joint T/NT approach can solve the resource scarceness problem of traditional terrestrial edge computing techniques. Recently several authors have highlighted the importance of aerial edge computing facilities for boosting the VNs performance [18], [19].

The recently introduced IoV technology supports the transmission, storage, and computing of huge amount of data generated by VTs, which can be used to improve the VNs performance [20]. Various ML-based approaches can be adapted for finding useful patterns from VTs data and to enable the intelligence into VNs. In [21], authors have surveyed various ML techniques, their applications, and challenges faced during their implementations over the dynamic vehicular networks. Recent works highlight the importance of FL for vehicular cases. In [11], authors have surveyed the various opportunities and challenges while considering FL over a federated VN. Though FL brings advantages in terms of communication efficiency and privacy preservation, additional optimization is needed in terms of device selection, resource allocation, distribution of the learning process, etc for adapting it over wireless environments. In [14], the authors have considered the FL device selection problem over a resource-constrained VN. A min-max optimization problem is formed and solved through a greedy algorithm. In [5], the authors have considered the joint computation offloading and FL process optimization over the edge computing enabled VN. Various cluster-based and distributed approaches are considered for finding the proper resource sharing between two phases aimed at minimizing the latency and energy costs. The importance of using FL in VN scenarios is also enforced by considering other aspects. Indeed, even if not strictly related to the

system under consideration, FL is a good candidate when jointly used with Blockchain for implementing a collaborative intrusion detection solution for IoV scenarios [22]. Given the rise of different security threats over edge environments, e.g., edge device compromise, privacy leaks, denial of service (DoS), additional security measures are needed while accessing the services from the edge facilities. In [23], the authors have proposed a lightweight anonymous mutual authentication scheme for  $n$ -times computation offloading in IoT environments. The proposed method can provide user anonymity, conditional message tracing, unlinkability to users' private data through messages collected from open channels, resilience to edge compromise and DoS. Similarly, in [24], the authors have proposed a two-factor lightweight privacy-preserving authentication scheme for enhancing the security of vehicular communication systems by using decentralization of central authorities and biological-password-based two-factor authentication. For another important IoT use case, the authors in [25] proposed a cloud-based user authentication scheme for secure authentication of medical data for wearable healthcare monitoring systems. The proposed scheme also allows password change, smart card revocation, and new wearable sensor addition phases.

In the case of future intelligent networks, various AI-based services can be enabled through proper collaboration between different networking environments and cloud/edge facilities [26]. However, this gives rise to several data privacy and security challenges [27]. FL, being one of the potential distributed learning techniques, can be useful for providing such AI-driven services in vehicular networks. Though FL has a main advantage in terms of elevated data privacy compared to the traditional centralized training approaches, in recent times various new privacy and security-related issues have risen in the traditional centralized FL models. Data/model poisoning, data modification, attacks on inference processes, backdoor attacks, Generative Adversarial Network-based attacks, malicious servers, free-riding attacks, and eavesdropping are some of the major security and/or privacy threats that can be seen while implementing the FL process over different IoT networks [28]. While considering the FL process over a vehicular system, this issue can even become more critical mainly due to dynamicity, presence of a large number of VTs, multiple server nodes, high sensitivity of vehicular data, and fatal impacts of a data breach, etc. Several techniques introduced over different IoT environments for enabling the secure and trustworthy FL process such as reputation management, Blockchain-based systems, data privacy-based perturbation techniques, secure aggregation techniques, secure multi-party computation, homomorphic encryption, back-door defenders, etc., need further analysis for creating a highly reliable FL over VNs [29], [30], [31].

In recent times, the edge intelligence paradigm along with various distributed learning frameworks have received lots of attention [32]. In [33], authors have proposed a federated RL approach for minimizing the communication delay of the traditional centralized training approach for finding proper offloading decisions and the resource allocation in a UAV enabled MEC environment. In another case, [13] highlights the importance of the distributed FL processing for solving

the long-distance connectivity and energy efficiency challenges of traditional centralized FL. MEC-enabled aerial access networks and their benefits for FL are discussed in [34]. In [35], the authors have proposed a communication-efficient FL framework based on a customized local training strategy, partial client participation, and flexible aggregation strategies. The analysis is limited to the terrestrial RSU nodes along with the cloud facilities.

Though FL has received lots of attention, a proper communication efficient, and sustainable FL platform for dynamic VN applications is still not functional. In addition, while designing the FL models, several authors have restricted their studies to terrestrial networks. By considering these shortcomings and various challenges posed by the FL, in this work, we aim to design the communication efficient, highly sustainable, distributed FL process for vehicular applications.

### III. SYSTEM MODEL AND PROBLEM FORMULATION

In the following, an urban Internet of Vehicles (IoV) scenario for Intelligent Transportation Systems with connected and intelligent VTs is considered, allowing to request several intelligent services from the nearby edge computing facilities. In recent times, such urban IoV scenarios have gained a lot of attention from the vehicular research community [14], [36]. In particular, we consider a multi-layered joint air-ground network composed of HAPs, UAVs (i.e., LAP nodes), RSUs, deployed along the road paths, and randomly distributed VTs traveling on a road in either directions, where  $\mathcal{V} = \{v_1, \dots, v_m, \dots, v_M\}$ ,  $\mathcal{R} = \{r_1, \dots, r_n, \dots, r_N\}$ ,  $\mathcal{U} = \{u_1, \dots, u_l, \dots, u_L\}$ , correspond to the sets denoting  $M$  VTs,  $N$  RSUs and  $L$  UAVs, respectively. Each HAP node is denoted through the index  $h$ .

The system is modeled in a time-discrete manner, and the network parameters are constant in each time interval  $\tau$ , where  $\tau_i$  identifies the  $i$ th time interval, i.e.,  $\tau_i = \{\forall t | t \in [i\tau, (i+1)\tau]\}$ . The generic  $m$ th VT is characterized by a processing capacity equal to  $c_{v,m}$  Floating Point Operations per Second (FLOPS) per CPU cycle, while its CPU frequency is  $f_{v,m}$  [5], [37]. VTs are supposed to be able to communicate on a bandwidth  $B_{v,m}^{\text{RSU}}$  with the RSUs, in a bandwidth  $B_{v,m}^{\text{LAP}}$  with UAVs and on a bandwidth  $B_{v,m}^{\text{HAP}}$  with the HAPs. In addition, the  $m$ th VT is supposed to hold a set  $\mathcal{D}_{v_m}$  with  $|\mathcal{D}_{v_m}| = \mathcal{K}_{v_m}$  data samples produced during its operation as a result of the embedded Advanced Driver-Assistance System (ADAS), and later used during the FL training process. FL is here exploited for assisting during vehicle operations, e.g., computation offloading, path planning, object detection.

The  $n$ th RSU, supposed to be in a fixed position with a coverage radius  $R_{r,n}$ , is characterized by a processing capacity equal to  $c_{r,n}$  FLOPS per CPU cycle, with CPU frequency  $f_{r,n}$ , and communication capabilities, supposed to be identified through a communication technology, able to cover the VTs on ground with an overall bandwidth  $B_{r,n}^{r \rightarrow v}$ . The RSUs are also able to connect with UAVs and the HAPs with a bandwidth  $B_{r,n}^{\text{UAV}}$  and  $B_{r,n}^{\text{HAP}}$ , respectively. The RSUs are connected to the electrical grid for the energy supply. Each RSU can provide edge computing services to the VTs in its coverage space.



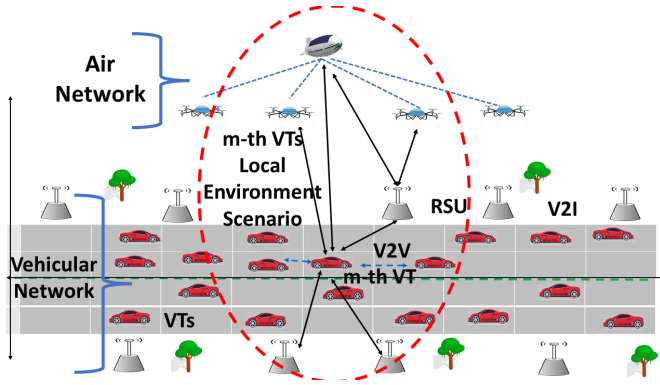


Fig. 1. The T/NT Integrated Scenario.

In addition, the area is supposed to be under the coverage of multiple UAVs with  $l$ th UAV at altitude  $\bar{h}_{u,l}$  and coverage radius  $R_{u,l}$ . We assume that UAVs are charged by exploiting available charging points in the service area. Based on VTs requests, UAVs can move in different directions with optimal path planning, whose management is beyond the scope of this work. While serving VTs, the  $l$ th UAV, supposed to move with a relatively slow speed compared with highly mobile VTs, is characterized by a processing capability equal to  $c_{u,l}$  FLOPS per CPU cycle, with CPU frequency  $f_{u,l}$ . In addition, it is supposed to be able to communicate on a bandwidth  $B_{u,l}^{l \rightarrow (v,r)}$  and cover an area with radius  $R_{u,l}$ , while the  $l$ th UAV has a bandwidth  $B_{u,l}^{\text{HAP}}$  when communicating with the HAP. Each UAV can serve a set of VTs and RSUs in its coverage space.

The generic  $h$ th HAP node is placed at an altitude  $h_h$  above the ground, and characterized by a processing capability  $c_h$  FLOPS per CPU cycle, with CPU frequency  $f_h$ . Moreover, we consider multi-beam antenna forming techniques, where each antenna beam is supposed to cover a geographical area of radius  $R_h$  and has a communication bandwidth  $B_h^{h \rightarrow (v,r,l)}$ . In the following, we will refer to a single beam as the coverage of the HAP. It should be noted that though HAP coverage is reduced to a single beam for notation simplicity, our approach can easily be scaled for the overall HAP coverage with multiple beams. Each RSU, UAV, and HAP provides edge computing services to the VTs, RSUs and UAVs within its coverage area. Fig. 1 shows the basic system elements and various communication links between them.<sup>1</sup>

#### A. VT Mobility Model

We suppose that the  $m$ th VT moves in a freeway-like mobility scenario with a speed  $\bar{v}_m(\tau_i)$  bounded by  $\bar{v}_{\min}$  and  $\bar{v}_{\max}$  [14], where the instantaneous speed is modeled through a truncated normal distribution density function:

$$f(\bar{v}_m(\tau_i)) = \begin{cases} \frac{2 \cdot \exp\left(-\frac{(\bar{v}_m(\tau_i) - \mu)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi} \left(\text{erf}\left(\frac{\bar{v}_{\max} - \mu}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{\bar{v}_{\min} - \mu}{\sigma\sqrt{2}}\right)\right)}, & \bar{v}_{\min} \leq \bar{v}_m(\tau_i) \leq \bar{v}_{\max} \\ 0, & \text{else} \end{cases} \quad (1)$$

<sup>1</sup>Despite the system model and the analysis is carried out considering the general case of multiple HAPs, the performance will be later evaluated for the simple case with only one HAP. The generic case can be seen as a simple extension of the one HAP scenario.

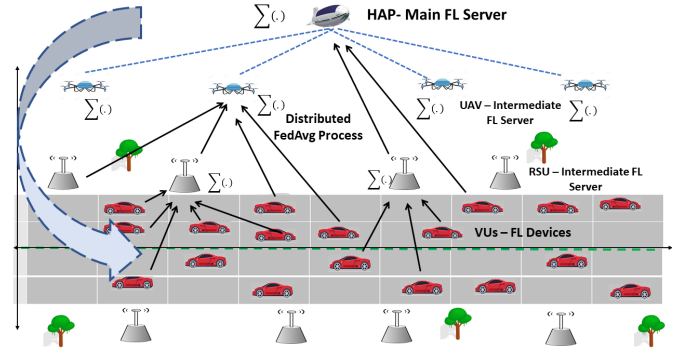


Fig. 2. Distributed FL Platform.

and  $\mu$  and  $\sigma$  are the mean and standard deviation of the vehicle's speed, and  $\text{erf}(x)$  is the Gauss error function over  $x$ . The path length within which the  $m$ th VT remains under the coverage of  $j$ th node (i.e., any of RSUs, UAVs or HAPs) is  $D_{v_m,j}(\tau_i)$  and can be given by:

$$D_{v_m,j}(\tau_i) = \sqrt{d_j^2 - (y_j - y_{v_m}(\tau_i))^2} \pm (x_j - x_{v_m}(\tau_i))$$

where,  $(x_{v_m}(\tau_i), y_{v_m}(\tau_i))$  is the location of the  $m$ th VT at  $\tau_i$  and  $(x_j, y_j)$  is the projection over the ground of a generic  $j$ th edge computing node, which can be a RSU, UAV or HAP. The available sojourn time for the  $m$ th VT with respect to a generic  $j$ th node can be written as:

$$T_{v_m,j}^{\text{soj}}(\tau_i) = \frac{D_{v_m,j}(\tau_i)}{|\bar{v}_m(\tau_i)|} \quad \forall j$$

#### B. Distributed FL Platform for Vehicular Applications

In order to solve the VN management through the proposed air-ground network architecture, we propose a VTs service-based distributed FL platform (Fig. 2).

The federated training operation depends on the service request  $v$ , where  $v$  can be any vehicular service requested by VTs, such as computation offloading towards edge servers, path planning, streaming-related services, etc. Each service  $v$  requires a unique FL model  $\mathcal{F}_v$ . In the considered FL platform, VTs (i.e., FL client devices) with local datasets  $\mathcal{D}_{v_m}$  can perform the local training for the FL model based upon the requested service  $v$ . Since different VTs can request different services over time, a group of VTs randomly located in the coverage space of the HAP requesting the same service  $v$  will participate collaboratively to train the FL model corresponding to the service  $v$ . The number of VTs participating in the training process of the  $v$ th FL model is given by:

$$M_v^{\text{FL}} = \{v_m | v_m \Leftrightarrow v, v_m \in M\} \quad \text{with } M_v^{\text{FL}} \subseteq M$$

In each  $i$ th FL iteration, after the local training operation, we assume that a data vector  $w_{v,m}^{it,v}(\tau_i)$  (i.e., model updates embedded into IP packet) is generated where an information header is added indicating the VTs service ( $\chi_v \Leftrightarrow v$ ). Here,  $\chi_v$  can be a unique sequence of bits indicating the  $v$ th service.<sup>2</sup>

<sup>2</sup>Though it is beyond the scope of this article,  $\chi_v$  can easily be modeled into a more sophisticated vector that can further enhance the security of the FL platform to tackle the various security/privacy related challenges discussed in the related works section.

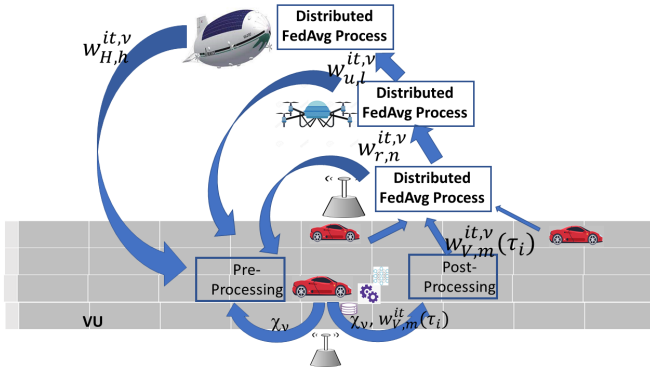


Fig. 3. Distributed FL Process.

Such processed data will be sent towards an upper layer edge computing node based upon the network selection strategy embraced by the VTs, as will be discussed in the following.

After receiving the data from the lower layer entities, each Edge Node (EN) will perform the FedAvg process creating the new set of updates  $w_{r,n}^{it,v}/w_{u,l}^{it,v}/w_h^{it,v}$ , where  $w_{r,n}^{it,v}$  is the aggregated FL model updates associated with the  $v$ th service generated by the intermediate RSU node  $n$ , and  $w_{u,l}^{it,v}$  and  $w_h^{it,v}$  are the FL model updates after the averaging process (i.e., FedAvg) performed at  $l$ th UAV node and  $h$ th HAP, respectively. With post-processing operation, the header information  $\chi_v$  is again inserted into the aggregated data ( $w_{r,n}^{it,v}/w_{u,l}^{it,v}/w_h^{it,v}$ ) for the next layer processing. In the end, data vector corresponding to the  $w_{r,n}^{it,v}/w_{u,l}^{it,v}/w_h^{it,v}$  is transmitted towards the next platform or VTs based upon the network selection strategy. Though it is beyond the scope of this work, insertion/processing of the header information associated with the specific service request allows the proposed FL platform to train multiple service-based FL models simultaneously.

After receiving data from the ENs, VTs use them in the next iteration of the FL process. The process continues for several FL iterations until a certain confidence interval is reached. Fig. 3 shows the steps of each single iteration of the proposed distributed FL process.

### C. Network Selection Parameters

FL performance is a function of the number of participating VTs to the FL process, the number of FL iterations performed by VTs, the communication and computing latency, and the energy cost of each FL iteration. The FL process cost depends also on the network selection strategy adopted by different networking layers given their limited computing and communication resources.

To better clarify this point, if a VT selects the HAP node direct link for the FL data transmission, it can potentially save the processing latency and the cost required to perform the FedAvg process at the intermediate layers; however, it can increase the VTs data transmission cost in terms of transmission latency and energy, mainly due to the limited resources of VTs and the long-distance communication links between VT and HAP. Also, due to long-distance communication links, the link failure probability can be higher, resulting in a possibly high number of dropouts (i.e., VTs not

participating in the FL training process). On the other hand, if VT decides to select RSU or UAV nodes for distributed FL data communication, it can potentially save communication time and energy. However, an additional burden of processing latency over these intermediate layers needs to be considered. Similar analysis can be applied to the RSU and UAV nodes when selecting the possible higher networking layers for the data communication. Therefore, there is a clear tradeoff between the different network selection strategies adapted by the VTs and the intermediate layers. A proper network selection strategy guaranteeing the optimal training latency and energy performance is required.

Based on their limited coverage ranges, each VT can be covered by set of RSUs, UAVs, and one HAP node. Focusing on the  $m$ th VT,  $1 \leq N_{v,m}^R \leq R_{max}$ ,  $1 \leq N_{v,m}^U \leq U_{max}$ , and  $1 \leq N_{v,m}^H \leq H_{max}$  represent the number of RSUs, UAVs and HAP nodes available for selection, respectively. Without loss of generality, we assume that  $N_{v,m}^R$ ,  $N_{v,m}^U$ , and  $N_{v,m}^H$  are lower bounded by 1 (i.e., every VT can be covered by at least one RSU, UAV and HAP), while  $R_{max}$ ,  $U_{max}$ , and  $H_{max}$  are the upper bounds on RSUs, UAVs, and HAP nodes covering it. Here, we define the following three decision variables modeling the network selection behavior of VTs, RSUs, and UAVs.

1) *VTs Network Selection Decision*: For the case of  $m$ th VT, we define

$$\mathbf{a}_{v,m}(\tau_i) = \left[ (0, 1)_{(1 \times N_{v,m}^R)}, (0, 1)_{(1 \times N_{v,m}^U)}, (0, 1)_{(1 \times N_{v,m}^H)} \right]$$

with dimension  $1 \times (N_{v,m}^R + N_{v,m}^U + N_{v,m}^H)$  modeling the available nodes for selection. VT can either select RSU, UAV, or HAP for communicating the FL model parameter updates. If  $\mathbf{a}_{v,m}(\tau_i) = \{0\}_{(1 \times (N_{v,m}^R + N_{v,m}^U + N_{v,m}^H))}$ , the  $m$ th VT does not participate in the FL process. Also, for avoiding the additional complexity, we consider that each VT can be assigned to only one EN which can be RSU, UAV, or HAP during the FL process. Thus,

$$\sum \mathbf{a}_{v,m}(\tau_i) \leq 1 \quad (2)$$

2) *RSUs Network Selection Decision*: For the case of  $n$ th RSU, we define

$$\mathbf{b}_{r,n}(\tau_i) = \left[ (0, 1)_{(1 \times N_{r,n}^U)}, (0, 1)_{(1 \times N_{r,n}^H)} \right]$$

with dimension  $1 \times (N_{r,n}^U + N_{r,n}^H)$  modeling the available nodes for selection. RSU node can either select UAV, or HAP for communicating the FL model parameter updates. If  $\mathbf{b}_{r,n}(\tau_i) = \{0\}_{(1 \times (N_{r,n}^U + N_{r,n}^H))}$ , the  $n$ th RSU node does not communicate with higher layers and broadcasts back the model parameters towards VT. For avoiding the additional complexity we consider that each RSU can be assigned to only one EN which can be UAV, or HAP during the FL process. Thus,

$$\sum \mathbf{b}_{r,n}(\tau_i) \leq 1 \quad (3)$$

3) *UAVs Network Selection Decision*: For the case of  $l$ th UAV, we define

$$\mathbf{c}_{u,l}(\tau_i) = \left[ (0, 1)_{(1 \times N_{u,l}^H)} \right]$$

with dimension  $1 \times N_{u,l}^H$  modeling the available nodes for selection. UAV can select HAP for communicating the FL model parameter updates or broadcast back the results towards VTs. If  $\mathbf{c}_{u,l}(\tau_i) = \{0\}_{(1 \times N_{u,l}^H)}$ , the  $l$ th UAV broadcast back the model parameters towards VT. For avoiding the additional complexity we consider that each UAV can be assigned to only one EN during the FL process. Thus,

$$\sum \mathbf{c}_{u,l}(\tau_i) \leq 1 \quad (4)$$

#### D. FL Process Cost Analysis

In general, FL is an iterative learning process where each FL iteration includes several steps adding latency and energy costs. Local on-device ML model training, data communication between VTs and FL servers, pre- and post-processing of FL model data, FedAvg process performed at FL servers are the main steps involved during FL iteration. In the following, we analyze the latency and energy cost of each of these operations.

1) *FL Local Training Model*: The FL computation corresponds to the local training of the ML model based on the on-device dataset. In local device training, the  $m$ th VT with service request  $v$  has to compute the local parameter set  $w_{v,m}^{it}$  through the dataset having size  $\mathcal{K}_{v,m}$  data samples; if we assume that, for every iteration, the total number of FLOPs required for each data sample  $d$  is  $\psi_d$ , the time and energy consumed during the FL training process by the  $m$ th device is [37]:

$$T_{v_m}^{\text{FL},c} = \frac{\sum_{d=1}^{\mathcal{K}_{v_m}} \psi_d}{c_{v,m} f_{v,m}}, \quad E_{v_m}^{\text{FL},c} = P_{v,m}^c \cdot T_{v_m}^{\text{FL},c}.$$

where  $P_{v,m}^c$  is the power consumed by the  $m$ th VT for the data processing. We suppose for simplicity that the on-device FL processing time and energy is the same for every iteration

2) *FL Data Pre-/Post-Processing*: For each FL iteration, pre- and post-processing operations are performed for detecting and adding the header information  $\chi_v$  associated with the service ( $v$ ) requested by the vehicular nodes. The latency and energy of these operations are:

$$T_i^{\text{FL},\text{hp}} = T_i^{\text{FL},\text{pre}} + T_i^{\text{FL},\text{post}}, \quad E_i^{\text{FL},\text{hp}} = P_i^c \cdot T_i^{\text{FL},\text{hp}}.$$

where,  $T_i^{\text{FL},\text{pre}} = \frac{N_i \cdot \psi_{pre}}{c_i \cdot f_i}$  is the time required to detect and remove the header information from FL data at the  $i$ th node, function of its computation resources, the number of FLOPs required to process the FL data (i.e., model parameters embedded in the IP packets) from node  $i$  given as  $\psi_{pre}$  and a number of VTs/servers sending the updates towards the server  $i$  given by  $N_i$ . Also,  $T_i^{\text{FL},\text{post}} = \frac{\psi_{post}}{c_i \cdot f_i}$  is the post-processing operation time required to insert the header information on model update vectors with  $\psi_{post}$  being a number of FLOPs required to process the updated model data after FedAvg.

3) *FL FedAvg Process*: In the proposed FL infrastructure, intermediate FL server (i.e., RSUs, UAVs, HAP) perform the FedAvg process on the data received from any of the lower layers. The latency and the energy required to perform the FedAvg process is given by:

$$T_i^{\text{FL},\text{FA}} = \frac{N_i \cdot \psi_{FA}}{c_i \cdot f_i}, \quad E_i^{\text{FL},\text{FA}} = P_i^c \cdot T_i^{\text{FL},\text{FA}}$$

where  $\psi_{FA}$  is the number of FLOPs required to process the individual nodes parameter vectors over  $i$ th server

4) *FL Data Communication Model*: The data rate between  $i$ th and  $j$ th node is a function of the mutual distance, hence:

$$r_{i,j}^{it}(B_i, d_{i,j}) = B_i \log_2 \left( 1 + \frac{P_i^{\text{tx}} \cdot h(d_{i,j})}{N_0} \right) \quad (5)$$

where  $P_i^{\text{tx}}$  is the transmission power of the generic  $i$ th device,  $h(d_{i,j})$  is the channel gain at a distance  $d_{i,j}$  between the  $i$ th device and the  $j$ th device, and  $N_0 = N_T B_i$  is the noise power, where  $N_T$  and  $B_i$  are the noise power spectral density and bandwidth associated to the  $i$ th device during communication.

During the FL processing, at each iteration  $it$ , the  $i$ th FL-device sends the parameters set  $w_i^{it}$  to the higher layers. Supposing that  $|w_i^{it}|$  represents the data size of the parameters set expressed in bits [10], the uplink transmission time and energy for the FL parameters in the  $it$ th iteration is:

$$T_{ij,it}^{\text{FL},\text{tx}} = \frac{|w_i^{it}|}{r_{i,j}^{it}(B_i^j, d_{i,j})}, \quad E_{ij,it}^{\text{FL},\text{tx}} = P_i^{\text{tx}} \cdot T_{ij,it}^{\text{FL},\text{tx}},$$

where,  $r_{i,j}^{it}$  is the uplink transmission rate between  $i$ th and the  $j$ th FL node, during the  $it$ th iteration, which is a function of the bandwidth ( $B_{i,m}^j$ ), and the distance ( $d_{i,j}$ ) between the two nodes, modeled through the Shannon capacity formula in (5). Since FL-servers are accessed by multiple VTs/lower layer nodes, we assume for simplicity that the  $j$ th node bandwidth is equally shared among the connected VTs and lower layer nodes, i.e., if  $u_j = u_l \in \mathcal{U}$ , the bandwidth resources of  $u_l$ ,  $B_{u,l}^{l \rightarrow (v,r)}$  is shared among all VTs and RSUs connected to it. Also,  $P_i^{\text{tx}}$  is the  $i$ th device transmission power. Similarly, the reception time required to receive data from the  $j$ th node by the  $i$ th node is given by,

$$T_{ij,it}^{\text{FL},\text{rx}} = \frac{|w_j^{it}|}{r_{i,j}^{it}(B_i^j, d_{i,j})}, \quad E_{ij,it}^{\text{FL},\text{rx}} = P_i^{\text{rx}} \cdot T_{ij,it}^{\text{FL},\text{rx}},$$

Each FL server needs to wait for receiving the data from all the connected VTs and lower layer nodes before performing the FedAvg process. The data reception latency and energy at the  $j$ th FL server are given by:

$$T_{j,it}^{\text{FL},\text{rx}} = \max_i \left\{ T_{i,it}^{\text{FL},\text{tx}} \right\}, \quad E_{j,it}^{\text{FL},\text{rx}} = \sum_i P_j^{\text{rx}} \cdot T_{j,it}^{\text{FL},\text{rx}},$$

With these basic latency and energy elements in hand, we can now define the FL iteration cost in terms of total latency and energy requirements.

#### E. FL Iteration Cost

The  $m$ th VT FL process cost includes the local computation cost, header processing operation cost at VT and the additional cost depending on the network selection strategy. Thus, for the  $m$ th VT, the total FL process cost (in terms of latency and

energy consumed) for a single iteration is:

$$\begin{aligned}
& T_{v,m,it}^{FL}(\mathbf{a}_{v,m}(\tau_i), \mathbf{b}_{r,n}(\tau_i), \mathbf{c}_{u,l}(\tau_i)) \\
& = T_{v_m}^{FL,c} + T_{v_m}^{FL,hp} + \mathbf{a}_{v,m}(\tau_i) \\
& \quad \times \begin{bmatrix} \left[ T_{v_m,r_n,it}^{FL,tx} + T_{r,n,it}^{FL}(\mathbf{b}_{r,n}(\tau_i), \mathbf{c}_{u,l}(\tau_i)) \right]_{(N_{v,m}^R \times 1)} \\ \left[ T_{v_m,u_l,it}^{FL,tx} + T_{u,l,it}^{FL}(\mathbf{c}_{u,l}(\tau_i)) \right]_{(N_{v,m}^U \times 1)} \\ \left[ T_{v_m,h,it}^{FL,tx} + T_{h,it}^{FL} \right]_{(N_{v,m}^H \times 1)} \end{bmatrix} \\
& E_{v,m,it}^{FL}(\mathbf{a}_{v,m}(\tau_i), \mathbf{b}_{r,n}(\tau_i), \mathbf{c}_{u,l}(\tau_i)) \\
& = E_{v_m}^{FL,c} + E_{v_m}^{FL,hp} + \mathbf{a}_{v,m}(\tau_i) \\
& \quad \times \begin{bmatrix} \left[ E_{v_m,r_n,it}^{FL,tx} + E_{r,n,it}^{FL}(\mathbf{b}_{r,n}(\tau_i), \mathbf{c}_{u,l}(\tau_i)) \right]_{(N_{v,m}^R \times 1)} \\ \left[ E_{v_m,u_l,it}^{FL,tx} + E_{u,l,it}^{FL}(\mathbf{c}_{u,l}(\tau_i)) \right]_{(N_{v,m}^U \times 1)} \\ \left[ E_{v_m,h,it}^{FL,tx} + E_{h,it}^{FL} \right]_{(N_{v,m}^H \times 1)} \end{bmatrix}
\end{aligned}$$

where, for the  $n$ -th RSU, the FL process cost for a single iteration is a function of the time/energy required to receive model updates from VTs, the header processing cost, the FedAvg process cost, and the additional cost based upon the network selection strategy adopted by it. Thus, for the case of  $n$ th RSU,

$$\begin{aligned}
& T_{r,n,it}^{FL}(\mathbf{b}_{r,n}(\tau_i), \mathbf{c}_{u,l}(\tau_i)) \\
& = T_{r_n,it}^{FL,rx} + T_n^{FL,hp} + T_{r_n}^{FL,FA} + \mathbf{b}_{r,n}(\tau_i) \\
& \quad \times \begin{bmatrix} \left[ T_{r_n,u_l,it}^{FL,tx} + T_{u,l,it}^{FL}(\mathbf{c}_{u,l}(\tau_i)) \right]_{(N_{r,n}^U \times 1)} \\ \left[ T_{r_n,h,it}^{FL,tx} + T_{h,it}^{FL} \right]_{(N_{r,n}^H \times 1)} \end{bmatrix} \\
& \quad + \left(1 - \sum \mathbf{b}_{r,n}(\tau_i)\right) T_{r,v,it}^{FL,tx} \\
& E_{r,n,it}^{FL}(\mathbf{b}_{r,n}(\tau_i), \mathbf{c}_{u,l}(\tau_i)) \\
& = E_{r_n,it}^{FL,rx} + E_{r_n}^{FL,hp} + E_{r_n}^{FL,FA} + \mathbf{b}_{r,n}(\tau_i) \\
& \quad \times \begin{bmatrix} \left[ E_{r_n,u_l,it}^{FL,tx} + E_{u,l,it}^{FL}(\mathbf{c}_{u,l}(\tau_i)) \right]_{(N_{r,n}^U \times 1)} \\ \left[ E_{r_n,h,it}^{FL,tx} + E_{h,it}^{FL} \right]_{(N_{r,n}^H \times 1)} \end{bmatrix} \\
& \quad + \left(1 - \sum \mathbf{b}_{r,n}(\tau_i)\right) E_{r,v,it}^{FL,tx}
\end{aligned}$$

Similarly, for the  $l$ -th UAV, the FL process cost is based upon data reception, header processing, FedAvg process, and the additional cost due to the network selection strategy. Thus, for the case of  $l$ th UAV, a single iteration cost is,

$$\begin{aligned}
& T_{u,l,it}^{FL}(\mathbf{c}_{u,l}(\tau_i)) = T_{u_l,it}^{FL,rx} + T_{u_l}^{FL,hp} + T_{u_l}^{FL,FA} \\
& \quad + \mathbf{c}_{u,l}(\tau_i) \times \left[ \left( T_{u_l,h,it}^{FL,tx} + T_{h,it}^{FL} \right)_{(N_{u,l}^H \times 1)} \right] \\
& \quad + \left(1 - \sum \mathbf{c}_{u,l}(\tau_i)\right) \cdot T_{l,v,it}^{FL,tx} \\
& E_{u,l,it}^{FL}(\mathbf{c}_{u,l}(\tau_i)) = E_{u_l,it}^{FL,rx} + E_{u_l}^{FL,hp} + E_{u_l}^{FL,FA} \\
& \quad + \mathbf{c}_{u,l}(\tau_i) \cdot \left[ \left( E_{u_l,h,it}^{FL,tx} + E_{h,it}^{FL} \right)_{(N_{u,l}^H \times 1)} \right] \\
& \quad + \left(1 - \sum \mathbf{c}_{u,l}(\tau_i)\right) \cdot E_{l,v,it}^{FL,tx}.
\end{aligned}$$

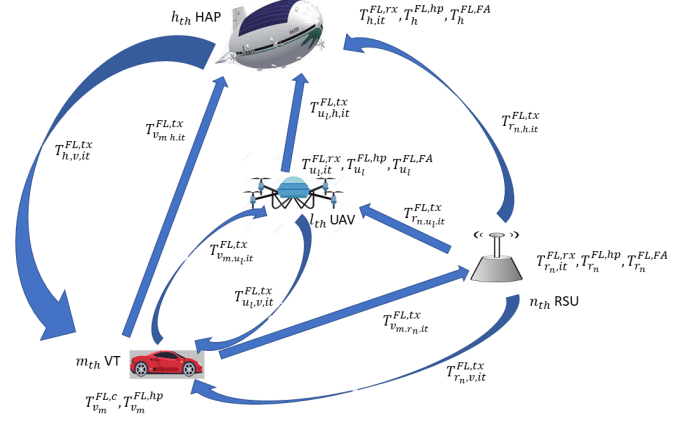


Fig. 4. Distributed FL Process Latency Analysis.

Finally, for the HAP node, the FL process cost for a single iteration is:

$$\begin{aligned}
T_{h,it}^{FL} & = T_{h,it}^{FL,rx} + T_h^{FL,hp} + T_h^{FL,FA} + T_{h,v_m,it}^{FL,tx} \\
E_{h,it}^{FL} & = E_{h,it}^{FL,rx} + E_h^{FL,hp} + E_h^{FL,FA} + E_{h,v_m,it}^{FL,tx}.
\end{aligned}$$

Fig. 4, presents the different latency components considered during the modeling of the FL latency over different nodes. We have avoided including the energy elements for simplicity.

In the end, for each FL iteration the required latency and energy cost for the  $m$ th VT is given by<sup>3</sup>:

$$\begin{aligned}
T_{it}^{FL}(d(v_m, r_n, u_l, \tau_i)) & = \max_m \{ T_{v,m,it}^{FL}(d(v_m, r_n, u_l, \tau_i)) \} \\
E_{it}^{FL}(d(v_m, r_n, u_l, \tau_i)) & = E_{v,m,it}^{FL}(d(v_m, r_n, u_l, \tau_i))
\end{aligned}$$

#### F. Number of FL Iterations Performed

Each FL iteration adds cost in terms of required latency and energy consumed over different platforms. However, it is important to perform a sufficient number of FL iterations for generating the FL model with sufficient accuracy over the real world data. The number of FL iterations performed by VTs depends on the adopted network selection strategy and the sojourn time within each EN coverage area. It is supposed that each VT can participate in the FL process till it belongs to the considered ENs coverage area. Thus,

$$\rho(d(v_m, r_n, u_l, \tau_i)) \leq \frac{T_{v_m,j}^{soj}(\tau_i)}{T_{it}^{FL}(d(v_m, r_n, u_l, \tau_i))} \quad (6)$$

where,  $\rho(d(v_m, r_n, u_l, \tau_i))$  is the number of FL iterations performed by the  $m$ th VT whose value is upper bounded by the ratio between the  $j$ th ENs sojourn time,  $T_{v_m,r_n}^{soj}(\tau_i)$ , and the FL iteration time. Here, the  $j$ th node corresponds to any RSU, UAV or HAP based upon the network selection strategy adapted by the  $m$ th VT. It should be noted that the  $j$ th node corresponds to the FL server node that transmit back the global model parameters towards the VT.

In general the FL process can be stopped if it achieves some predefined stopping criteria, such as the number of FL iterations performed, predefined loss function value,

<sup>3</sup>For notational simplicity hereafter, we use  $d(v_m, r_n, u_l, \tau_i)$  as a decision vector notation indicating the three decision vectors  $\mathbf{a}_{v,m}(\tau_i)$ ,  $\mathbf{b}_{r,n}(\tau_i)$ , and  $\mathbf{c}_{u,l}(\tau_i)$  together.



etc. [5], [38]. Therefore, without loss of generality, we introduce  $\epsilon_v$  as a convergence parameter in terms of FL global model loss function value, for the FL model corresponding to the service  $v$ . In the past, it has been shown that, in certain environments, it is possible to limit the number of FL iterations required to be performed to achieve the predefined loss function value [5], [39], [40]. However, the maximum number of FL iterations required to be performed can depend upon several parameters such as local environment scenarios, number of VTs participating in the training process, quality of VTs data, etc. Here, we assume that the number of FL iterations required to achieve the FL performance is

$$\rho_v^{\max} = \frac{\mathcal{C}}{\sqrt{M_v^{FL}}} \quad (7)$$

function of the number of VTs participating in the training process of the FL model of the  $v$ th service, where the square-root models the reduced impact when a higher number of VTs participate to the FL training process. Here,  $\mathcal{C}$  is a constant representing the maximum number of iterations required for a single VT to achieve the FL model convergence. If VTs have participated to a reduced number of FL iterations when using the FL model for their applications, the performance can be sub-optimal. In such cases VTs might need to pay additional penalty in terms of performance degradation or reduced quality of service.

Here, we introduce a stochastic penalty function  $\mathcal{P}_v^{\text{FL}}(\rho(d(v_m, r_n, u_l, \tau_i)))$  for measuring the impact of the number of FL iterations performed over an FL model performance. This analysis is motivated by the work done in [5], for the joint computation offloading and the FL process optimization over VN. If the  $m$ th VT requesting the service  $v$  is using the FL process to estimate the parameter  $x_v$  with  $x_{\min, v} \leq x_v \leq x_{\max, v}$ , and the estimated value is given by  $\hat{x}_v(\rho(d(v_m, r_n, u_l, \tau_i)))$ , the FL penalty is:

$$\begin{aligned} \mathcal{P}_v^{\text{FL}}(\rho(d(v_m, r_n, u_l, \tau_i))) \\ = \sqrt{(x_v - \hat{x}_v(\rho(d(v_m, r_n, u_l, \tau_i))))^2} \end{aligned} \quad (8)$$

where  $\hat{x}_v$  is estimated by using a stochastic function with truncated normal distribution with probability density function  $f_{\hat{x}_v}(\cdot)$  of  $\hat{x}_v$  as,

$$f_{\hat{x}_v}(\hat{x}_v; \bar{\mu}, \bar{\sigma}) = \begin{cases} \frac{1}{\bar{\sigma}} \frac{\xi\left(\frac{\hat{x}_v - \bar{\mu}}{\bar{\sigma}}\right)}{\bar{\Delta}\left(\frac{x_{\max, v} - \bar{\mu}}{\bar{\sigma}}\right) - \bar{\Delta}\left(\frac{x_{\min, v} - \bar{\mu}}{\bar{\sigma}}\right)} & \text{if } x_{\min, v} \leq \hat{x}_v \leq x_{\max, v} \\ 0 & \text{otherwise} \end{cases}$$

and  $\xi(\cdot)$  and  $\bar{\Delta}(\cdot)$  are, respectively, the probability density function of the related standard normal distribution and its cumulative distribution function, i.e.,

$$\xi(\omega) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\omega^2}{2}}, \quad \bar{\Delta}(\kappa) = \frac{1}{2} \left[ 1 + \operatorname{erf}\left(\frac{\kappa}{\sqrt{2}}\right) \right].$$

In this work we assume that the mean value of the distribution of  $x$ , i.e.,  $\bar{\mu}$ , and its variance,  $\bar{\sigma}^2$ , are equal to

$$\bar{\mu} = x_v, \quad \bar{\sigma}^2 = \left[ \bar{\gamma} \cdot \frac{\rho_v^{\max} - \rho(d(v_m, r_n, u_l, \tau_i))}{\rho_v^{\max}} \right]^2$$

where  $\bar{\gamma}$  is a numerical constant, used for controlling the variance of the model. The interested reader can have a look to [5] where the same authors considered the above model for estimating the FL iterations vs performance for the computation offloading application over VN.

In the end the total FL latency and energy cost is:

$$\begin{aligned} T^{\text{FL}}(d(v_m, r_n, u_l, \tau_i)) &= \rho T_{it}^{\text{FL}}(d(v_m, r_n, u_l, \tau_i)) \\ E^{\text{FL}}(d(v_m, r_n, u_l, \tau_i)) &= \rho E_{it}^{\text{FL}}(d(v_m, r_n, u_l, \tau_i)) \end{aligned}$$

### G. Problem Formulation

In this work, we aim to perform a communication-efficient FL process over a joint air-ground network. By adopting a proper network selection strategy over different platforms ( $\mathcal{A} = \{d(v_m, r_n, u_l, \tau_i)\}, \forall m, n, l$ ), the aim is to maximize the FL process performance. Thus the main aim is to minimize the joint cost of latency, energy, and the penalty function value measuring the FL process performance:

$$\begin{aligned} \mathbf{P1} : \mathcal{A}^* = \operatorname{argmin}_{\mathcal{A}} \left\{ \frac{1}{M_v^{FL}} \sum_{m=1}^{M_v^{FL}} (\eta_1 T^{\text{FL}}(d(v_m, r_n, u_l, \tau_i)) \right. \\ \left. + \eta_2 E^{\text{FL}}(d(v_m, r_n, u_l, \tau_i)) \right. \\ \left. + w_1 \mathcal{P}_v^{\text{FL}}(\rho(d(v_m, r_n, u_l, \tau_i))) \right\} \end{aligned} \quad (9)$$

subject to the following constraints,

$$\mathbf{C1} : \text{Eq. (2), (3), (4)} \quad (10a)$$

$$\mathbf{C2} : \text{Eq. (6)} \quad (10b)$$

$$\mathbf{C3} : \text{Eq. } \sum B_i^j \leq B^j \quad \forall j \in \mathcal{R}, \mathcal{U}, h \quad (10c)$$

$$\mathbf{C4} : 0 \leq \eta_1, \eta_2 \leq 1; \quad \eta_1 + \eta_2 = 1, w_1 \geq 0 \quad (10d)$$

where  $\mathcal{A} = \{d(v_m, r_n, u_l, \tau_i)\}$  is the combined set of network selection decisions of all nodes involved during the FL process,  $\eta_1$  and  $\eta_2$  are weighting coefficients for balancing latency and energy consumption, and  $w_1$  is a weighting coefficient for the penalty function. According to (10a), each VT, RSU and UAV can communicate with only one EN from the upper layers. Eq. (10b) limits the number of iterations performed by each VT depending on the available sojourn time considering the limited HAP coverage. Eq. (10c) shows the upper limit on the bandwidth resources of any  $j$ th EN, among any RSU, UAV or HAP. The total bandwidth available for the VTs and other nodes connected to any  $j$ th EN will be upper bounded by the bandwidth of the  $j$ th node, i.e.,  $B^j$ . According to (10d), weighting coefficients  $\eta_1$  and  $\eta_2$  can have any value between zero and one with their sum equal to one. Also  $w_1$  can have any positive value.

## IV. PROPOSED SOLUTIONS

For solving (9), we aim at finding a proper EN selection strategy for creating a highly reliable FL model with reduced latency and energy costs. With multiple edge computing layers and a large number of VTs along the road, the considered problem can be hard to solve. Here, we propose a MDP-based RL approach for finding proper assignment strategies for different nodes. The basic elements of the MDP



model include the state space ( $\mathcal{S}$ ), the action space ( $\mathcal{A}$ ), the reward function ( $\mathcal{R}$ ), the discount factor ( $\gamma$ ), and the proper environment dynamics, or state transition, probability model ( $\mathcal{P}$ ). Thus, the MDP process can be defined as a tuple given by  $\{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma\}$ . In order to analyze the performance of the proposed MDP model, we present two multi-dimensional MDP approaches based on the VTs' local environment, as well three benchmark methods for comparison purposes.

#### A. MDP Based Approach

1) *Local Environment Based Multi-Dimensional MDP Model*: In the considered network architecture, each VT can be covered by one or more RSUs, UAVs, and one HAP. Thus, different VTs/ENs can have different number of nodes available for communicating the FL updates. In order to properly select the EN for performing the FL, and setting up the MDP parameters, we assume that each VT is able to acquire the local environment parameters through V2X communication links, allowing more personalized MDP models with better accuracy. In particular, we classify the VTs into different groups, based on their local environments, where each group can have a separate state space and action space.

For the  $m$ th VT, the main parameters include the number of RSUs ( $N_{v,m}^R$ ), UAVs ( $N_{v,m}^U$ ), and HAP ( $N_{v,m}^H$ ) nodes available for the FL process. In addition, we define three vectors  $V_{v_m}^R = \left\{ V_{v_m, r_1}^R, \dots, V_{v_m, r_{(N_{v,m}^R)}}^R \right\}$  with  $V_{v_m, r_n}^R \leq V_{max}^R$ ,  $V_{v_m}^U = \left\{ V_{v_m, u_1}^U, \dots, V_{v_m, u_{(N_{v,m}^U)}}^U \right\}$  with  $V_{v_m, u_l}^U \leq V_{max}^U$ , and  $V_{v_m}^H = \left\{ V_{v_m, h_1}^H, \dots, V_{v_m, h_{(N_{v,m}^H)}}^H \right\}$  with  $V_{v_m, h}^H \leq V_{max}^H$ , corresponding to the number of nodes (i.e., VTs, RSUs, and UAVs) already connected to the each RSU, UAV and HAP node, respectively, covering the  $m$ th VT. Here,  $V_{max}^R$ ,  $V_{max}^U$  and  $V_{max}^H$  stand for the maximum number of devices that can be served by each RSU, UAV and HAP nodes. Thus, a tuple  $\kappa = \{N_{v,m}^R, N_{v,m}^U, N_{v,m}^H, V_{v_m}^R, V_{v_m}^U, V_{v_m}^H\}$  can represent the  $m$ th VT local environment. The number of possible  $\kappa$  values, i.e.,  $\bar{K}$ , can depend upon  $V_{max}^R$ ,  $V_{max}^U$  and  $V_{max}^H$ . Through V2X communication links, VTs can determine the number of nodes around them. However, since all VTs participate in the FL process simultaneously, their assignment parameters in advance is unknown. Therefore, some assumptions are required. Here, we consider the following two approaches for generating  $V_{v_m}^R$ ,  $V_{v_m}^U$ , and  $V_{v_m}^H$  vectors that can be used to improve the MDP models accuracy.

a) *Minimum distance based assignment approach*: In the case of a minimum distance-based approach, each node is assigned to the upper layer node with the minimum possible distance. For example,  $m$ th VT is assigned to the nearest RSU,  $n$ th RSU is assigned to the nearest UAV, and  $l$ th UAV is assigned to the nearest HAP. Thus in general,

$$a_{v,m}(\tau_i) = 1 \iff n = \underset{n \in N_{v,m}^R}{\operatorname{argmin}} \{d_{v_m, r_n}(\tau_i)\} \quad (11a)$$

$$b_{r,n}(\tau_i) = 1 \iff l = \underset{l \in N_{v,m}^U}{\operatorname{argmin}} \{d_{r_n, u_l}(\tau_i)\} \quad (11b)$$

$$c_{u,l}(\tau_i) = 1 \iff h = \underset{h \in N_{v,m}^H}{\operatorname{argmin}} \{d_{u_l, h}(\tau_i)\} \quad (11c)$$

b) *Random assignment approach*: In this approach, each node is assigned to any of the higher layer nodes with a probabilistic rule. We have considered the uniform assignment approach where the probability of assigning the  $i$ th node towards the  $j$ th upper layer node is given by

$$p(i \rightarrow j) = \frac{1}{U_i^{max}} \quad (12)$$

where  $U_i^{max}$  indicate the total number of upper layer nodes covering the  $i$ th node which can be VT, RSU or UAV.

With these two approaches in hand, different sets of  $V_{v_m}^R$ ,  $V_{v_m}^U$ , and  $V_{v_m}^H$  can be generated, helping to select proper ENs. The two different MDP approaches resulting from these methods are denoted as MDP with minimum distance based assignment approach (MDP-MD), and MDP with random assignment approach (MDP-RA). Later, the performance of these two schemes is compared in the simulation results section.

2) *State Space ( $\mathcal{S}$ )*: In general, MDP state space is constituted by all possible states in which MDP agents can find themselves during the exploration of the environment. Finding an appropriate network of ENs, i.e., vehicle to HAP ( $v_m \rightarrow h$ ), vehicle to  $n$ th RSU to HAP ( $v_m \rightarrow r_n \rightarrow h$ ), vehicle to  $l$ th UAV to HAP ( $v_m \rightarrow u_l \rightarrow h$ ), vehicle to  $n$ th RSU to  $l$ th UAV to HAP ( $v_m \rightarrow r_n \rightarrow u_l \rightarrow h$ ) can potentially save the FL iteration latency and energy cost and allow VTs to participate to a large number of FL iterations resulting into a better FL model generation. In this work,  $\mathcal{S}$  is constituted by multiple number of binary variables corresponding to all  $n \in N_{v,m}^R$ ,  $l \in N_{v,m}^U$  and  $h$ . In particular, we define

$$S_R^n(d(v_m, r_n, u_l, \tau_i)) = \begin{cases} 1 & \text{if } v_m \rightarrow r_n \\ & \text{and } \frac{\rho(d(v_m, r_n, u_l, \tau_i))}{\rho_v^{max}} > \zeta_\rho^R \\ 0 & \text{otherwise} \end{cases}$$

as a binary variable related to  $r_n \in N_m^R$ , which takes value 1 if the  $m$ th VT is assigned to the  $n$ th RSU and able to perform a sufficient number of FL iterations, where  $0 < \zeta_\rho^R \leq 1$  is a parameter indicating the FL accuracy level that can be based on the service type requested by the users. For example, in case of a critical safety-related service, the FL model accuracy should be high for avoiding possible fatal car crashes due to the failure of the FL models. In this case,  $\zeta_\rho^R$  should be closer to 1 or even 1. On the other hand, if it is not a high priority/safety-related service, a moderate FL accuracy can be sufficient to serve the user. In such cases  $\zeta_\rho^R$  can be smaller. Similarly,

$$S_U^l(d(v_m, r_n, u_l, \tau_i)) = \begin{cases} 1 & \text{if } (v_m \rightarrow u_l \text{ or } v_m \rightarrow r_n \rightarrow u_l) \\ & \text{and } \frac{\rho(d(v_m, r_n, u_l, \tau_i))}{\rho_v^{max}} > \zeta_\rho^U \\ 0 & \text{otherwise} \end{cases}$$

is a binary variable related to  $u_l \in N_m^U$ , which takes 1 if the  $m$ th VT is assigned to the  $l$ th UAV and able to perform a sufficient number of FL iterations. Here,  $0 < \zeta_\rho^U \leq 1$  is the parameter indicating the FL accuracy level as a function of

the service type requested by the users. Finally,

$$S_H^h(d(v_m, r_n, u_l, \tau_i)) = \begin{cases} 1 & \text{if } (v_m \rightarrow h \text{ or } v_m \rightarrow r_n \rightarrow h \text{ or } v_m \rightarrow u_l \rightarrow h, \\ & \text{or } v_m \rightarrow r_n \rightarrow u_l \rightarrow h) \text{ and } \frac{\rho(d(v_m, r_n, u_l, \tau_i))}{\rho_v^{\max}} \\ & > \zeta_\rho^H \\ 0 & \text{otherwise} \end{cases}$$

is a binary variable related to  $h$ , which takes value 1 if the  $m$ th VT is assigned to the HAP  $h$  and able to perform a sufficient number of FL iterations. Here,  $0 < \zeta_\rho^H \leq 1$  is the parameter indicating the FL accuracy level based upon the service type requested by the users.

In the end, if the  $m$ th VTs local environment is modeled through the tuple  $\kappa$ , the complete state vector is given as,

$$S_\kappa = \left\{ S_R^1, \dots, S_R^{N_{v,m}^R}, S_U^1, \dots, S_U^{N_m^U}, S_H^h \right\}$$

3) *Action Space ( $\mathcal{A}$ ):* If the  $m$ th VT local environment is modeled through the tuple  $\kappa$ , the action space ( $\mathcal{A}_\kappa = \{a_\kappa(\tau_i)\}$ ) includes all possible actions  $a_\kappa$  that can be taken by the MDP agent corresponding to the  $\kappa$ . In the considered FL network selection problem, agents can select ENs belonging to the different networking layers. Therefore, the generic action space is defined as,

$$\mathcal{A}_\kappa = \left\{ (v_m \rightarrow r_n), (v_m \rightarrow u_l), (v_m \rightarrow h), (v_m \rightarrow r_n \rightarrow u_l), (v_m \rightarrow u_l \rightarrow h), (v_m \rightarrow r_n \rightarrow h), (v_m \rightarrow r_n \rightarrow u_l \rightarrow h) \right\} \forall m, l \quad (13)$$

4) *Reward Function ( $\mathcal{R}$ ):* MDP agents can receive a positive or negative reward based upon the current state and the action taken. Here, we consider the weighted sum of latency, energy and penalty functions cost required to complete a single FL iteration as a reward received by the agent based upon its state and action. Thus,

$$\mathcal{R}_{v,\kappa}(s, a) = \eta_1 T^{\text{FL}}(s_\kappa, a_\kappa) + \eta_2 E^{\text{FL}}(s_\kappa, a_\kappa) + w_1 \mathcal{P}_v^{\text{FL}}(\rho(s_\kappa, a_\kappa))$$

5) *MDP Environment Dynamics ( $\mathcal{P}$ ):* MDP environment dynamics model the behavior of the MDP environment in terms of state transition probabilities based upon the agents' current state and the actions performed. The probability of MDP agent finding itself into state  $s'$  when it performs the action  $a$  from state  $s$  is given as  $P(s'|s, a)$ . Modeling such state transition probability over dynamic vehicular environments can be challenging. We propose a time-dependent state transition probability equation based upon the MDP agent's local environment. In general, for scenario  $\kappa$ , the state transition probability at  $\tau_i$  is given by

$$P(s_\kappa(\tau + \delta) | s_\kappa(\tau), a_\kappa(\tau)) = P\left(\left\{S_R^{r_n}(\tau + \delta), S_U^{u_l}(\tau + \delta), S_H^h(\tau + \delta)\right\} \times \left| \left\{S_R^{r_n}(\tau), S_U^{u_l}(\tau), S_H^h(\tau)\right\}, a_\kappa(\tau)\right.\right)$$

which represents the state transition probability for state  $s_\kappa(\tau + \delta)$  for the MDP agent from current state  $s_\kappa(\tau)$  taking action  $a_\kappa(\tau)$ . Here,  $\delta$  is the MDP time step. Since VT can connect to

only one node in a given time interval, the events  $S_R^{r_n}$ ,  $S_U^{u_l}$  and  $S_H^h$  can be considered as an independent events, which results into,

$$P(s_\kappa(\tau + \delta) | s_\kappa(\tau), a_\kappa(\tau)) = P\left(S_R^{r_n}(\tau + \delta) \mid \left\{S_R^{r_n}(\tau), S_U^{u_l}(\tau), S_H^h(\tau)\right\}, a_\kappa(\tau)\right) \cdot P\left(S_U^{u_l}(\tau + \delta) \mid \left\{S_R^{r_n}(\tau), S_U^{u_l}(\tau), S_H^h(\tau)\right\}, a_\kappa(\tau)\right) \cdot P\left(S_H^h(\tau + \delta) \mid \left\{S_R^{r_n}(\tau), S_U^{u_l}(\tau), S_H^h(\tau)\right\}, a_\kappa(\tau)\right) \quad (14)$$

In particular, with various communication links, e.g., V2V, V2R, V2I, vehicular-based MDP agents can acquire useful information about the surrounding environment (i.e., tuple  $\kappa$ ), which can be used to model the state transition probabilities. The transition probability expressions are modeled as exponential functions based upon various local environment parameters. The state transition probability expressions for  $S_R^{r_n}$  for the  $\kappa$ th MDP agent with current state  $s_\kappa(\tau)$  and performing action  $a_\kappa(\tau)$  is defined as,

$$P(\{S_R^{r_n}(\tau + \delta) = 1\} | s_\kappa(\tau), a_\kappa(\tau)) = \begin{cases} \exp(-\lambda_n^R(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow r_n \\ 0 & \text{else} \end{cases} \quad (15)$$

$$P(\{S_R^{r_n}(\tau + \delta) = 0\} | s_\kappa(\tau), a_\kappa(\tau)) = 1 - P(\{S_R^{r_n}(\tau + \delta) = 1\} | s_\kappa(\tau), a_\kappa(\tau)) \quad (16)$$

corresponding, respectively, to the probability that the MDP agent will be in state with  $S_R^{r_n}(\tau + \delta) = 1$  and  $S_R^{r_n}(\tau + \delta) = 0$  by taking action  $a_\kappa(\tau)$  from current state  $s_\kappa(\tau)$ . Also,  $\lambda_n(\tau_i) = \delta_1^R \cdot V_{v_m, r_n}^R(\tau_i) + \delta_2^R \cdot d_{v_m, r_n}(\tau_i) + \delta_3^R / T_{v_m, r_n}^{soj}(\tau_i)$  models the impact of VTs local environment over the state transition probability values. According to  $\lambda_n(\tau_i)$ , if the  $m$ th VT through action  $a_\kappa(\tau)$  selects the RSU  $n$  with high  $V_{v_m, r_n}^R$  and  $d_{v_m, r_n}(\tau_i)$ , the VTs might not be able to perform the required number of FL iterations. Also if VT selects the RSU with a high sojourn time value, it can perform a sufficient number of FL iterations, resulting in a higher probability that occurs  $S_R^{r_n}(\tau + \delta) = 1$ ;  $\delta_1^R$ ,  $\delta_2^R$  and  $\delta_3^R$  are the weighing coefficients used to associate proper weights towards each parameters. Next, for  $S_U^{u_l}$ ,

$$P(\{S_U^{u_l}(\tau + \delta) = 1\} | s_\kappa(\tau), a_\kappa(\tau)) = \begin{cases} \exp(-\lambda_{l,1}^U(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow u_l \\ \exp(-\lambda_{l,2}^U(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow r_n \rightarrow u_l \\ 0 & \text{else} \end{cases}$$

$$P(\{S_U^{u_l}(\tau + \delta) = 0\} | s_\kappa(\tau), a_\kappa(\tau)) = 1 - P(\{S_U^{u_l}(\tau + \delta) = 1\} | s_\kappa(\tau), a_\kappa(\tau))$$

corresponding to the probabilities that MDP agent in state  $s_\kappa(\tau)$  by taking action  $a_\kappa(\tau)$  will find itself in state with  $S_U^{u_l}(\tau + \delta) = 1$  and  $S_U^{u_l}(\tau + \delta) = 0$ , respectively.

Since VT can be connected to the UAV directly, or through RSU as an intermediate node, two separated cases are provided for increasing the accuracy of the MDP framework. Here,  $\lambda_{l,1}^U(\tau_i) = \delta_1^U V_{v_m, u_l}^U(\tau_i) + \delta_2^U d_{v_m, u_l}(\tau_i) + \delta_3^U / T_{v_m, u_l}^{soj}(\tau_i)$  and  $\lambda_{l,2}^U(\tau_i) = \delta_4^U V_{v_m, r_n}^R(\tau_i) + \delta_1^U V_{v_m, u_l}^U(\tau_i) + \delta_5^U (d_{v_m, r_n}(\tau_i) + d_{r_n, u_l}(\tau_i)) + \delta_3^U / T_{v_m, u_l}^{soj}(\tau_i)$  are the two parameters measuring the impact of surrounding environment over the transition

probabilities, while  $\delta_1^U, \dots, \delta_5^U$  correspond to the weighting coefficients for properly balancing the impact of various environment parameters. In the end, for  $S_H^h$ :

$$P \left( \left\{ S_H^h(\tau + \delta) = 1 \right\} \middle| s_\kappa(\tau), a_\kappa(\tau) \right) = \begin{cases} \exp(-\lambda_{h,1}^H(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow h \\ \exp(-\lambda_{h,2}^H(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow r_n \rightarrow h \\ \exp(-\lambda_{h,3}^H(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow u_l \rightarrow h \\ \exp(-\lambda_{h,4}^H(\tau_i)) & \text{if } a_\kappa(\tau) \in v_m \rightarrow r_n \rightarrow u_l \rightarrow h \\ 0 & \text{else} \end{cases} \quad (17)$$

$$P \left( \left\{ S_H^h(\tau + \delta) = 0 \right\} \middle| s_\kappa(\tau), a_\kappa(\tau) \right) = 1 - P \left( \left\{ S_H^h(\tau + \delta) = 1 \right\} \middle| s_\kappa(\tau), a_\kappa(\tau) \right) \quad (18)$$

are the state transition probabilities corresponding to  $S_H^h(\tau + \delta) = 1$  and  $S_H^h(\tau + \delta) = 0$ , respectively. Since VTs can communicate with the HAP node through various links, different cases are presented based upon the action taken. In (17),

$$\begin{aligned} \lambda_{h,1}^H(\tau_i) &= \delta_1^H V_{v_m,h}^H(\tau_i) + \delta_2^H d_{v_m,h}(\tau_i) + \delta_3^H / T_{v_m,h}^{soj}(\tau_i) \\ \lambda_{h,2}^H(\tau_i) &= \delta_4^H V_{v_m,r_n}^R(\tau_i) + \delta_1^H V_{v_m,h}^H(\tau_i) \\ &\quad + \delta_5^H (d_{v_m,r_n}(\tau_i) + d_{r_n,h}(\tau_i)) + \delta_3^H / T_{v_m,h}^{soj}(\tau_i) \\ \lambda_{h,3}^H(\tau_i) &= \delta_6^H V_{v_m,u_l}^U(\tau_i) + \delta_1^H V_{v_m,h}^H(\tau_i) \\ &\quad + \delta_7^H (d_{v_m,u_l}(\tau_i) + d_{u_l,h}(\tau_i)) + \delta_3^H / T_{v_m,h}^{soj}(\tau_i) \\ \lambda_{h,4}^H(\tau_i) &= \delta_4^H V_{v_m,r_n}^R(\tau_i) + \delta_6^H V_{v_m,u_l}^U(\tau_i) + \delta_1^H V_{v_m,h}^H(\tau_i) \\ &\quad + \delta_8^H (d_{v_m,r_n}(\tau_i) + d_{r_n,u_l}(\tau_i) \\ &\quad + d_{u_l,h}(\tau_i)) + \delta_3^H / T_{v_m,h}^{soj}(\tau_i) \end{aligned}$$

are the parameters modeling the surrounding environments impact over the state transitions. In the end, by using (14), (18) can be used to find the state transition probability in any interval  $\tau$ .

### B. MDP-Based FL Network Selection Strategy

For the MDP model corresponding to the  $\kappa$ th agent, the solutions' set can be defined as a policy function  $\pi_\kappa = \{\pi_\kappa(s_\kappa(\tau_i + \delta)), \forall \delta\}$  that maps every state  $s_\kappa \in \mathcal{S}$  to action  $a_\kappa \in \mathcal{A}$ . Selecting different actions can result in different policy functions, where the aim is to find an optimal policy that corresponds to the minimum cost in terms of delay, energy and FL process penalty value. For every policy  $\pi_\kappa$ , a value function  $V_{\pi_\kappa}(s_\kappa(\tau_i))$ , corresponding to a state  $s_\kappa(\tau_i)$  can be defined for analyzing its performance. In general,  $V_{\pi_\kappa}(s_\kappa(\tau_i))$  corresponds to an expected value of a discounted sum of total reward received by following the policy  $\pi_\kappa$  from state  $s_\kappa(\tau_i)$ , and can be defined as:

$$V_{\pi_\kappa}(s_\kappa(\tau_i)) = \mathbb{E} \left\{ \sum_{\delta=0}^{\Delta} \gamma^\delta R(s_\kappa(\tau_i + \delta), \pi_\kappa(s_\kappa(\tau_i + \delta))) \right\}$$

where  $\gamma \in [0, 1]$  is the discount factor,  $R(s_\kappa(\tau_i + \delta), \pi_\kappa(s_\kappa(\tau_i + \delta)))$  is the immediate reward received for following a policy  $\pi_\kappa$  at time  $\tau_i + \delta$  from the state  $s_\kappa(\tau_i + \delta)$ ,  $\Delta$  is the maximum

number of steps considered during the MDP evaluation, i.e., episode length, and  $\mathbb{E}(\cdot)$  corresponds to the expected value. Thus, the value function analyzes the particular policy function by assigning a numeric value to each state and can be utilized to compare the performance of different policies. In the end, the following optimization problem can be formulated in order to be able to find the best possible policy function associated with state  $s_\kappa(\tau_i)$ :

$$V(s_\kappa(\tau_i)) = \min_{\pi_\kappa \in \Pi_\kappa} V_{\pi_\kappa}(s_\kappa(\tau_i)) \quad (19)$$

where  $\Pi_\kappa$  corresponds to the set of policy functions that can be explored.

As shown by many works (e.g., [41], [42]), the problem defined in (19), can converge into a Bellman optimality equation given by:

$$\begin{aligned} V(s_\kappa(\tau_i)) &= \min_{a_\kappa(\tau_i) \in \mathcal{A}_\kappa(\tau_i)} \left\{ R(s_\kappa(\tau_i), a_\kappa(\tau_i)) \right. \\ &\quad \left. + \gamma \sum_{s_\kappa(\tau_i + \delta) \in \mathcal{ST}} Pr \{s_\kappa(\tau_i + \delta) \mid s_\kappa(\tau_i), a_\kappa(\tau_i)\} V(s_\kappa(\tau_i + \delta)) \right\} \quad (20) \end{aligned}$$

Different approaches can be used to solve the problem in (20); however, the value iteration approach is widely known for its fast convergence and easy implementation. Therefore, below we present a value iteration approach aimed at solving the MDP designed in the previous section for finding an optimal policy that corresponds to the minimization of a FL process time and energy over VN.

The value iteration method allows finding an optimal policy and value function for the MDP models. The Algorithm 1 describes the steps involved during the value iteration process. For every agent  $\kappa$ , the process begins by initializing the values of each state to  $\infty$  and iteration count ( $it$ ) to 0 (Line 2). For each state-action pair, the state value is determined by using (21) (Line 5). The state value and a corresponding optimal policy ( $\pi_\kappa^*(s_\kappa(\tau_i))$ ) associated with state  $s_\kappa$  is determined by using (22) and (23) (Lines 7-8). The iterative process continues till the change in the all states values becomes less than the predefined convergence parameter  $\epsilon$  (Lines 10-13). In the end, the algorithm returns the set of optimal policy functions  $\{\pi_\kappa^*\}$  associated with all possible scenarios in which VTs can find themselves over the road (Line 15).

The time complexity of the traditional value iteration process can be analyzed as  $\mathcal{O}(\Delta|\mathcal{S}| \cdot |\mathcal{A}|)$  with  $\Delta$  being the maximum number of time steps considered,  $|\mathcal{S}|$  state space dimension, and  $|\mathcal{A}|$  representing the action space. With the involvement of  $\bar{K}$  scenarios, the time complexity expression becomes  $\mathcal{O}(\bar{K}\Delta|\mathcal{S}| \cdot |\mathcal{A}|)$ . The scenario-based modeling can reduce the state and action space dimensions significantly. Additionally, time-dependent state transition probabilities can reduce the overall uncertainty in the MDP process.

### C. Benchmark Methods

For analyzing the performance of the proposed MDP model, we have considered the following benchmark methods.

**Algorithm 1** MDP Value Iteration**Input:**  $\epsilon, \gamma, S_\kappa, A_\kappa, Pr, \bar{K}, \Delta$ **Output:**  $\{\pi_\kappa^*\}$ 


---

```

1: for  $\kappa \in \bar{K}$  do
2:   Initialize  $it = 0, V^0(s_\kappa(\tau_i)) = \infty, \forall s_\kappa(\tau_i)$ 
3:   for  $s_\kappa(\tau_i) \in S_\kappa$  do
4:     for  $a_\kappa(\tau_i) \in A_\kappa$  do
5:
6:        $V^{it+1}(s_\kappa(\tau_i), a_\kappa(\tau_i))$ 
7:        $\leftarrow R(s_\kappa(\tau_i), a_\kappa(\tau_i))$ 
8:        $+ \gamma \sum_{s_\kappa(\tau_i+\delta) \in S_\kappa} Pr(s_\kappa(\tau_i+\delta) | s_\kappa(\tau_i), a_\kappa(\tau_i)) v^{it}(s_\kappa(\tau_i+\delta))$ 
9:     end for
10:     $V^{it+1}(s_\kappa(\tau_i)) = \min_{a_\kappa(\tau_i)} V^{it+1}(s_\kappa(\tau_i), a_\kappa(\tau_i))$ 
11:
12:     $\pi_\kappa^*(s_\kappa(\tau_i)) = \operatorname{argmin}_{a_\kappa(\tau_i)} V^{it+1}(s_\kappa(\tau_i), a_\kappa(\tau_i))$ 
13:
14:  end for
15:  if  $|v^{it+1}(s_\kappa(\tau_i)) - v^{it}(s_\kappa(\tau_i))| > \epsilon$  then
16:     $it = it + 1$ 
17:  else
18:     $\pi_\kappa^* = \{\pi_\kappa^*(s_\kappa(\tau_i))\}$ 
19:  end if
20: end for
21: return  $\{\pi_\kappa^*\}$ 

```

---

1) *Conventional Centralized FL Process (C-FL)*: In the case of a conventional centralized FL process, each VT transmits its model updates to the centralized HAP server. Thus,  $v_m \rightarrow h, \forall v \in \mathcal{V}$ . This approach can reduce the overall processing costs in terms of intermediate layer processing and averaging operations performed over RSUs and UAVs. However, possible long-distance communication links between VTs and HAP can limit the performance in terms of link failures, high energy costs, limited users participating in the FL process, etc.

2) *Minimum Distance Based FL Process (MD-FL)*: In this case, the FL process assumes that each node communicates with the nearest nodes from the upper layer. Thus, each participating VT can select the shortest distance RSU node for transmitting its update, which then process and transmit the aggregated update vectors towards the nearest UAV for further processing. In the end, HAP collects data from all the participating UAV terminals for generating the global model, which it then broadcasts back towards VTs. Eqs. (11a)-(11c) can be used to determine the minimum distance assignment vectors for different nodes.

3) *Random Assignment Based FL Process (RA-FL)*: In this approach, nodes involved in the FL process (i.e., VTs, RSUs, UAVs, and HAP) follow the random assignment strategy

in (12). Thus, each VT selects the one EN from a set of RSUs, UAVs, and HAP covering it. Similarly, RSUs can either be connected to the UAV/HAP or can also communicate back the results to VTs. UAVs also followed the same strategy, where they can either send their data to HAP or return it to the VTs for the next round of the FL process.

4) *FedCPF Inspired RSU-Based Benchmark Solution for the Considered Scenario (FedR-FL)*: In [35], authors have proposed a FedCPF approach based upon a customized local training strategy, partial client participation, and flexible aggregation strategies. Here we considered a FedCPF-inspired, RSU-based benchmark approach where VTs are performing the local training process and transmitting the model parameters to the nearest RSU node. The client selection strategy of the FedCPF approach is considered where participation of each VT in the FL process is based upon a probability  $P_{sel}$  (i.e.,  $P_{sel}$  is the probability of the client being a part of FL training, while  $(1 - P_{sel})$  is the probability that the client will opt out from the LF training). The other two strategies of customized local training strategy and the deadline-based server aggregation strategy are based upon the RSU sojourn time constraint. In particular, VTs' participation in the FL process is limited by its dynamicity and the RSU coverage range.

## V. PERFORMANCE EVALUATION

The value iteration algorithm for solving the MDP model and the benchmark methods previously described are simulated over a Python-based simulator, using ML-related libraries such as NumPy, Pandas, Matplotlib. In Table I, the main simulation parameters are shown for the considered network architecture. The service area is under the coverage of one HAP, 20 UAVs and 40 RSUs. A variable number of VTs between 200 and 700 are considered, assuming that each one is requesting service  $v$  with a probability equal to 0.2. Each VT is traveling with a variable speed as modeled in (1) with  $\mu=10$  m/s and  $\sigma=1$ . The maximum number of FL iterations required to achieve the proper performance, as defined in (7), consider that  $\mathcal{C} = 1000$ . Also  $P_{sel} = 0.7$  is used for the FedCPF-inspired RSU-based benchmark solution approach. Each VT has a FL dataset of size  $|\mathcal{D}_{v_m}| = 5,000$  samples. During the FL training process,  $\psi_d = 1500$  and  $T_{i,v}^{FA} = 1$  ms. The maximum number of nodes covering any VT is given by  $R_{max} = 3, U_{max} = 2$  and  $H_{max} = 1$ . Additionally a maximum number of nodes served by each RSU  $v_{max}^R = 8$ , each UAV  $v_{max}^U = 16$ , and HAP  $v_{max}^H = 32$  are considered. With the multi-core processing hardware of ENs, these users can be grouped into different levels based on the number of cores.

The weighting coefficients used for modeling the transition probabilities are defined as,  $[\delta_1^R, \delta_2^R, \delta_3^R] = [0.025, 0.004, 0.5]$ ,  $[\delta_1^U, \delta_2^U, \delta_3^U, \delta_4^U, \delta_5^U] = [0.2, 0.0125, 0.1, 0.2, 0.005]$ ,  $[\delta_1^H, \delta_2^H, \delta_3^H, \delta_4^H, \delta_5^H, \delta_6^H, \delta_7^H, \delta_8^H] = [100, 0.4, 100, 200, 1.25, 100, 0.4, 1.25] \cdot 10^{-3}$ . The weighting coefficients have been defined so that different local environment parameters, defined in the transition probability equations (16)-(18), have a value range consistent among them during the MDP state evaluation. For the case of FL penalty function  $x = 0.5$  with  $0 \leq x \leq 1$  is used along with  $\bar{\lambda} = 0.3$ . Additionally,  $\zeta_\rho^R, \zeta_\rho^U$ , and  $\zeta_\rho^H$  are



TABLE I  
SIMULATION PARAMETERS

HAP Coverage ( $R_h$ )	1.2 km
UAV Coverage ( $R_{u,l}$ )	100 m
RSU Coverage ( $(R_{r,n})$ )	50 m
VT Computation Cap. ( $c_{v,m} \cdot f_{v,m}$ )	10 GFLOPS
RSU Computation Cap. ( $c_{r,n} \cdot f_{r,n}$ )	20 GFLOPS
UAV Computation Cap. ( $c_{u,l} \cdot f_{u,l}$ )	20 GFLOPS
HAP Computation Cap. ( $c_h \cdot f_h$ )	40 GFLOPS
HAP Altitude ( $h_h$ )	10 km
UAV Altitude ( $h_{u,l}$ )	1 km
HAP Bandwidth ( $B_h^{h \rightarrow (v,r,l)}$ )	250 MHz
UAV Bandwidth ( $B_{u,l}^{l \rightarrow (v,r)}$ )	75 MHz
RSU Bandwidth ( $B_{r,n}^{r \rightarrow v}$ )	25 MHz
VT Speed Range ( $(\bar{v}_{\min}, \bar{v}_{\max})$ )	(8 m/s, 14 m/s)
HAP Power ( $P_{t_h}, P_{r_h}$ )	(1.1, 0.9) W
UAV Power ( $P_{t_{c,u}}, P_{r_{c,u}}^l$ )	(1.2, 1) W
RSU Power ( $P_{t_{c,r}}, P_{r_{c,r}}^m$ )	(1.3, 1.2) W
VT Power ( $P_{c,v}^m, P_{t_{c,v}}, P_{r_{c,v}}^m$ )	(1.1, 1.5, 1.3) W
Noise Power ( $N_T$ )	-110 dbm [44]
FLOPs Required ( $\psi_{pre}, \psi_{post}, \psi_{FA}$ )	$10^4, 10^4, 10^5$ FLOPs
Weighting Coefficients ( $\eta_1, \eta_2, w_1$ )	(0.5, 0.5, 1)

set to 0.7. During value iteration process  $\gamma = 0.9$ ,  $\epsilon = 0.01$ ,  $\lambda = 0.1$  and episode length  $\Delta = 200$  are used.

#### A. Numerical Results

In the following, we present the main performance results including the FL cost, latency, energy, FL penalty, and the average number of FL iterations performed by VTs for different methods.

1) *FL Process Cost*: The main objective of this work is to jointly reduce the overall latency, energy, and FL penalty. Fig. 5 shows the performance in terms of FL overall cost for the MDP schemes and the benchmark methods previously presented. It can be observed that both MDP methods outperform the benchmark approaches as the number of VTs increases.

With a reduced number of VTs, with a fully distributed FL process, the MD-FL approach requires higher cost mainly due to several processing operations performed at different layers. On the other hand, a fully centralized C-FL method has reduced costs due to the presence of a limited number of VTs requesting the resources from the centralized HAP node. However, if the number of VTs is higher, the overall cost of the C-FL approach grows fast mainly due to the higher communication distances and the limited resources of a HAP node. With this, the C-FL cost becomes higher than the other benchmark methods with the increasing density of VTs. Similar effects can be seen later in the latency and energy plots shown in Figs. 6 and 7. The other two benchmark approaches (RA-FL and FedCPF-inspired method) have a slightly better performance mainly due to the reduced communication distances and reduced processing operations compared to the fully distributed MD-FL and a fully centralized C-FL approaches. However, the imperfect/static edge node selection without considering the local environment parameters and the available resources, the performance of the benchmark approaches compared to the proposed MDP solutions.

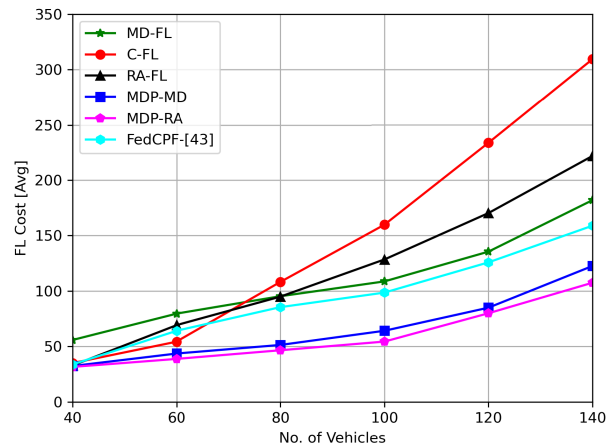


Fig. 5. Performance results in terms of overall cost function with variable number of active vehicles.

On the other hand, the proposed MDP solutions, with network selection based upon the VTs local environments and the available resources of ENs, are able to keep the FL process cost under the limit. In particular, the MDP-RA method outperforms all other approaches. For the case of the MDP-MD, the VTs local environment is modeled through the assignments of the FL devices to the nearest nodes with less flexibility, i.e., VTs can be assigned to the RSUs, RSUs can be assigned to the UAVs, and UAVs are assigned to the HAP node. On the other hand, the MDP-RA approach is more flexible, where each node can select any higher layer entities, i.e., VTs can be assigned to the RSU, UAVs, or HAP. Therefore, MDP-RA method outperforms the MDP-MD in terms of overall cost, as well FL latency, and energy, as later shown in Figs. 6 and 7.

With the upcoming latency-constrained vehicular applications and services demanding ML models with high accuracy, it is important to perform the distributed learning process, such as FL, in a limited time and with reduced energy consumption. Thus, the proposed distributed learning framework with efficient network selection strategies allows a large number of VTs to participate in the training process with reduced cost and a huge advantage over the traditional methods.

2) *FL Latency Performance*: For each FL iteration, the FL process is impacted by communication, training, and processing latency. In Fig. 6, we present the average FL iteration latency for various MDP and benchmark methods. In particular, MDP-MD and MDP-RA methods, with proper node selections can perform the FL process with reduced latency compared with other methods. As for the previous case, with fewer nodes requesting resources, the C-FL method performs better compared to the MD-FL and RA-FL methods. However, when VTs are more, C-FL performance in terms of latency requirements degrades drastically. On the other hand with a distributed FL process, the MD-FL approach induces higher latency with lower VTs. However, with higher VTs its performance is better than the C-FL method mainly due to the distribution of the FL process over the multiple edge nodes. The fedCPF-inspired approach selects the RSU nodes for the averaging operation limiting the latency costs in the beginning. However, with the limited sojourn time, VTs are unable to perform a sufficient number of iterations resulting in the higher

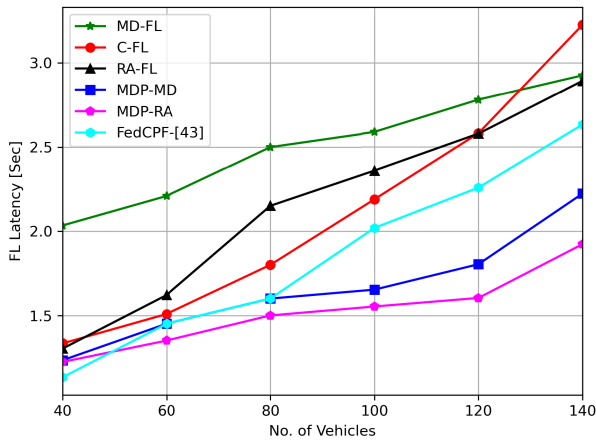


Fig. 6. FL latency with variable number of active vehicles.

FL penalty as shown later in Fig. 8. With a higher number of VTs, the performance of the FedCPF-inspired approach degrades mainly due to the limited RSU resources and the imperfect node selection.

The MDP approaches, especially the MDP-RA method, jointly reduce both communication and processing latency by distributing the FedAvg process over a sufficient number of edge nodes. Therefore, the proposed methods can efficiently train the FL model over the distributed multi-layered VN environments.

3) *FL Energy Performance*: It is important to reduce the FL process energy cost given the involvement of different T/NT networking platforms (i.e., VTs, UAVs) with scarce energy resources. The FL process consumes energy for the local training operations, data communication, and FL data processing over servers. In Fig. 7, we present the performance in terms of energy spent by the different methods. Similar to the latency performance, the C-FL process energy performance is better for a reduced number of VTs, due to the involvement of a limited number of VTs and reduced FedAvg process cost. However, when VTs are more, the VTs energy requirements become high mainly due to long-distance communication over limited bandwidth resources. Compared with the C-FL method, MD-FL has an advantage in terms of reduced communication distances/costs. However, with the repetition of the FedAvg process over each layer, the energy cost increases. On the other hand, the proposed MDP methods can reduce both communication and computation process energy requirements simultaneously by properly distributing the FL process over multi-layered VN. By utilizing the local environment knowledge, MDP methods can select proper ENs with sufficient resources and, as a result, are able to perform the FL process with reduced energy requirements.

It should also be noted that the energy performance of the C-FL method degrades quickly compared to its latency performance. This is mainly because every VTs involved in the FL process of the C-FL approach requires communication with the centralized HAP node. Due to this, the energy cost induced by the individual VTs can be higher compared to the other benchmark methods. This trend can also be seen in the overall cost performance in Fig. 5. The results from Figs. 6 and 7 can also highlight the issues of the well-known straggler effect in the FL framework with the traditional benchmark methods and

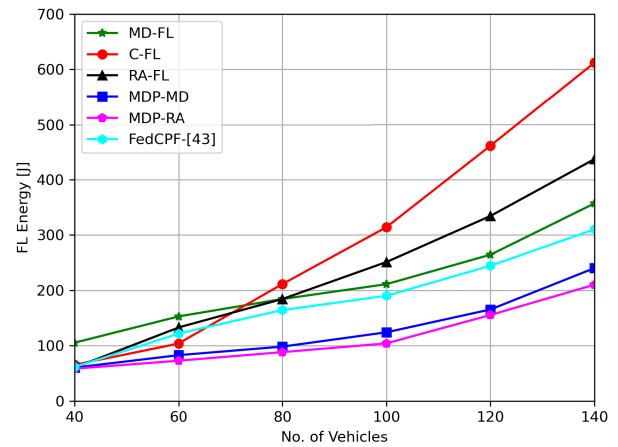


Fig. 7. Performance results in terms of energy consumption for the FL process with variable number of active vehicles.

the necessity to counter such effects with the new solutions. The proposed MDP-based methods can mitigate such effects as highlighted by the performance in Figs. 5-7.

4) *FL Penalty Performance*: With the adopted network selection strategy, if VTs fail to perform a sufficient number of FL iterations, the FL model performance may not be adequate. We have modeled the impact of the number of FL iterations performed by VTs in terms of a stochastic penalty function presented in (8). In Fig. 8, we show the average FL penalty value for different sets of VTs for the proposed methods. The benchmarks, with inadequate FL process, fail to perform the required number of FL iterations resulting in the higher FL penalties. The FedCPF-inspired approach selects the nearby RSU node for limiting the FL communication cost, which as a result limits the number of FL iterations performed by VTs, inducing the heavy FL penalty. The other two benchmark methods, MD-FL and RA-FL also suffer from a large penalty due to the reduced number of FL iterations performed mainly due to the high latency per FL iteration with constrained sojourn times. Although the C-FL method gains from a higher coverage range of the HAP node and with reduced FL latency, for a reduced number of VTs it is able to perform a large number of FL iterations with a reduced penalty, while, as the number of VTs increases, its performance decreases. For a reduced number of VTs, the penalty value for the MDP-RA process is high, mainly because of the low number of VTs participating in the FL process and its decisions to select the nearby edge nodes for reducing the overall FL cost. However, with a growing number of VTs, its performance increases with proper network selection strategies and an adequate number of VTs participating in the FL process. On the other hand, the MDP-MD method which suffers slightly in terms of latency and energy costs in the beginning can perform a high number of FL iterations reducing the LF penalty. Notice that these behaviors of MDP methods can also be impacted by the assumptions made over the competing VTs decisions and can have different impacts in terms of individual costs. However, both the MDP methods are able to reduce the joint costs of latency, energy and penalties significantly compare to the traditional benchmark methods. Therefore, the proposed FL process with proper network selections can create reliable FL models with better performance.

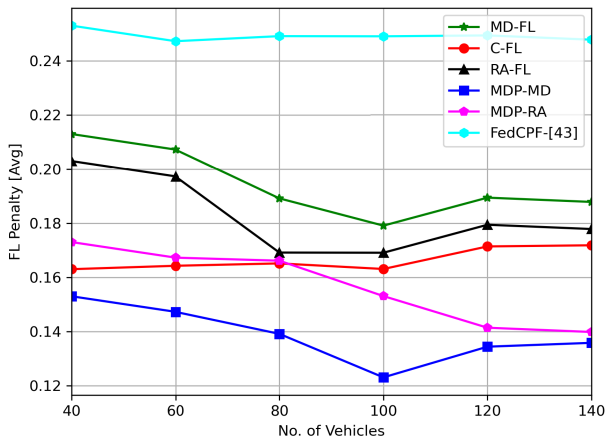


Fig. 8. Performance results in terms of FL Penalty value with variable number of active vehicles.

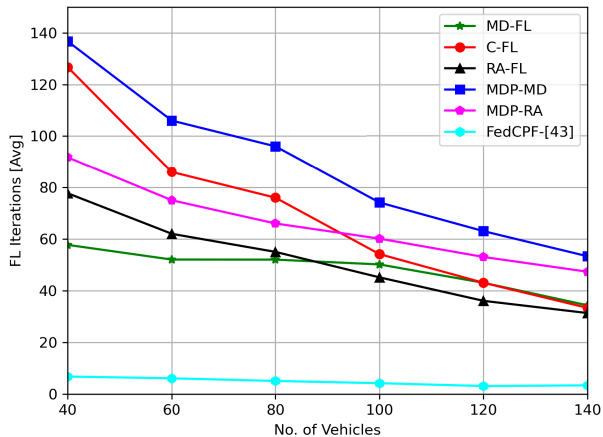


Fig. 9. Performance results in terms of average number of FL iterations with variable number of active vehicles.

5) *Average Number of FL Iterations*: For having adequate performance FL nodes, VTs should be able to perform a sufficient number of FL iterations ( $\rho(d(v_m, r_n, u_l, \tau_i))$ ). The number of FL iterations performed by each VT is based upon the network selection strategy and the available sojourn time of the selected ENs, as given in (6). A proper network selection strategy can reduce the FL iteration time. Also selecting proper ENs with a higher number of communication/computation resources allows VTs to participate in a larger number of iterations. To shade more light on the results presented in the previous figures, here we present the average number of FL iterations performed by different methods (Fig. 9). It can be seen that with a lower number of VTs, C-FL is able to perform a higher number of FL iterations, however, its performance reduces as more and more VTs participate in the process mainly due to the longer FL iteration time. It should also be noticed that though in the beginning, the C-FL approach can outperform one of the MDP solutions (MDP-RA), its joint performance is still not optimized due to the static FL process (Fig. 5). On the other hand, as described before in Fig. 8 the node selection strategies for the MDP-RA and MDP-MD methods are based upon a joint cost optimization and can be influenced by the competing VTs decisions. With FedCPF-inspired RSU-based benchmark solution, VTs can only perform a limited number of iterations only, mainly due to the limited coverage range of the RSU nodes. This also

highlights the importance of considering the distributed NTN layers of networking platforms for supporting the FL process. With imperfect edge node selection strategies, the other two benchmark solutions (MD-FL and RA-FL), also suffer from limited FL iterations resulting in imperfect FL models with higher performance penalties (i.e., Fig. 8).

## VI. CONCLUSION

In this work, we have presented the communication-efficient, distributed FL platform over a joint T/NT-based VN. The proposed approach can be useful for creating cost-efficient, sustainable, and more reliable FL models for serving VTs applications. With proper analysis of the FL process cost, we formed the constrained optimization problem for finding the optimal FL network selection strategy over multi-layered VNs. We further modeled the FL network selection problem as a sequential decision-making RL problem by adapting the MDP framework. A time-dependent environment dynamic model is created by utilizing the VTs environment parameters acquired through the V2X technology. In the end, the value iteration approach is used to solve the MDP model for finding suitable policies. The numerical results acquired over the Python-based simulation show the major advantages of the proposed FL approach over several other benchmark methods including the conventional centralized FL process. In the future, we expect to extend this work by analyzing the performance of proposed methods on realistic vehicular systems for enabling intelligent solutions at the edge.

## REFERENCES

- [1] P. Porombage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 1st Quart., 2018.
- [2] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020.
- [3] G. K. Kurt et al., "A vision and framework for the high altitude platform station (HAPS) networks of the future," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 729–779, 2nd Quart., 2021.
- [4] C. Xu and Z. Zhou, "Vehicular content delivery: A big data perspective," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 90–97, Feb. 2018.
- [5] S. Shinde, A. Bozorgchenani, D. Tarchi, and Q. Ni, "On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2041–2057, Feb. 2022.
- [6] A. Bozorgchenani, S. Maghsudi, D. Tarchi, and E. Hossain, "Computation offloading in heterogeneous vehicular edge networks: On-line and off-policy bandit solutions," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4233–4248, Dec. 2022.
- [7] Z. Ning, P. Dong, J. J. P. C. Rodrigues, F. Xia, and X. Wang, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–24, 2019.
- [8] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
- [9] W. Xia, W. Wen, K.-K. Wong, T. Q. S. Quek, J. Zhang, and H. Zhu, "Federated-learning-based client scheduling for low-latency wireless communications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 32–38, Apr. 2021.
- [10] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Oct. 2021.



- [11] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: Opportunities and solutions," *IEEE Netw.*, vol. 35, no. 2, pp. 152–159, Mar. 2021.
- [12] S. S. Shinde and D. Tarchi, "Towards a novel air-ground intelligent platform for vehicular networks: Technologies, scenarios, and challenges," *Smart Cities*, vol. 4, no. 4, pp. 1469–1495, Dec. 2021.
- [13] S. A. Khowaja, K. Dev, P. Khowaja, and P. Bellavista, "Toward energy-efficient distributed federated learning for 6G networks," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 34–40, Dec. 2021.
- [14] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11073–11087, Aug. 2022.
- [15] Z. Du, C. Wu, T. Yoshinaga, K.-L.-A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular Internet of Things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, 2020.
- [16] W. Zhan et al., "Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5449–5465, Jun. 2020.
- [17] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2005.
- [18] A. Traspadini, M. Giordani, and M. Zorzi, "UAV/HAP-assisted vehicular edge computing in 6G: Where and what to offload?" 2022, *arXiv:2202.10953*.
- [19] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 140–146, Feb. 2020.
- [20] W. Xu et al., "Internet of vehicles in big data era," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.
- [21] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 2027–2057, 3rd Quart., 2021.
- [22] H. Liu et al., "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6073–6084, Jun. 2021.
- [23] F. Wang, Y. Xu, L. Zhu, X. Du, and M. Guizani, "LAMANCO: A lightweight anonymous mutual authentication scheme for  $N$ -times computing offloading in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4462–4471, Jun. 2019.
- [24] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 896–911, Feb. 2016.
- [25] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 942–956, Sep. 2020.
- [26] Y. Xu et al., "Artificial intelligence: A powerful paradigm for scientific research," *Innovation*, vol. 2, no. 4, Nov. 2021, Art. no. 100179.
- [27] F. Wang, B. Diao, T. Sun, and Y. Xu, "Data security and privacy challenges of computing offloading in FINS," *IEEE Netw.*, vol. 34, no. 2, pp. 14–20, Mar. 2020.
- [28] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.
- [29] A. M. Elbir, B. Soner, S. Coleri, D. Gunduz, and M. Bennis, "Federated learning in vehicular networks," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Athens, Greece, Sep. 2022, pp. 72–77.
- [30] S. Otoum, I. Al Ridhawi, and H. T. Mouftah, "Blockchain-supported federated learning for trustworthy vehicular networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taiwan, Dec. 2020, pp. 1–6.
- [31] E. Muscinelli, S. S. Shinde, and D. Tarchi, "Overview of distributed machine learning techniques for 6G networks," *Algorithms*, vol. 15, no. 6, p. 210, Jun. 2022.
- [32] J. Park et al., "Communication-efficient and distributed learning over wireless networks: Principles and applications," *Proc. IEEE*, vol. 109, no. 5, pp. 796–819, May 2021.
- [33] Y. Nie, J. Zhao, F. Gao, and F. Yu, "Semi-distributed resource management in UAV-aided MEC systems: A multi-agent federated reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13162–13173, Dec. 2021.
- [34] Q.-V. Pham, M. Zeng, T. Huynh-The, Z. Han, and W.-J. Hwang, "Aerial access networks for federated learning: Applications and challenges," *IEEE Netw.*, vol. 36, no. 3, pp. 159–166, May 2022.
- [35] S. Liu, J. Yu, X. Deng, and S. Wan, "FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022.
- [36] S. S. Shinde and D. Tarchi, "Network selection and computation offloading in non-terrestrial network edge computing environments for vehicular applications," in *Proc. 11th Adv. Satell. Multimedia Syst. Conf. 17th Signal Process. Space Commun. Workshop (ASMS/SPSC)*, Graz, Austria, Sep. 2022.
- [37] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *J. Commun. Inf. Netw.*, vol. 6, no. 2, pp. 110–124, Jul. 2021.
- [38] M. Chen et al., "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.
- [39] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.
- [40] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 4, pp. 1078–1088, Dec. 2021.
- [41] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2483–2493, Feb. 2021.
- [42] X. Zhang, J. Zhang, Z. Liu, Q. Cui, X. Tao, and S. Wang, "MDP-based task offloading for vehicular edge computing under certain and uncertain transition probabilities," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3296–3309, Mar. 2020.
- [43] M. Khayyat, I. A. Elgendy, A. Muthanna, A. S. Alshahrani, S. Alharbi, and A. Koucheryavy, "Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks," *IEEE Access*, vol. 8, pp. 137052–137062, 2020.



**Swapnil Sadashiv Shinde** (Student Member, IEEE) was born in Pune, India, in 1991. He received the B.Eng. degree in electronics and telecommunication engineering from Pune University, India, in 2013, the M.Des. degree in communication systems from the Indian Institute of Information Technology, Design and Manufacturing, Kancheeppuram, India, in 2015, and the M.S. degree in telecommunication engineering from the University of Bologna, Italy, in 2020, where he is currently pursuing the Ph.D. degree in connected vehicles for beyond 5G scenarios.

From 2015 to 2017, he worked as a Project Engineer at the Indian Institute of Technology, Kanpur, India. His research interests include edge computing, reinforcement learning, distributed machine learning, and non-terrestrial networks.



**Daniele Tarchi** (Senior Member, IEEE) was born in Florence, Italy, in 1975. He received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in informatics and telecommunications engineering from the University of Florence, Florence, in 2000 and 2004, respectively.

From 2004 to 2010, he was a Research Associate with the University of Florence. From 2010 to 2019, he was an Assistant Professor at the University of Bologna, Bologna, Italy, where he is currently an Associate Professor. He is the author of around 150 published papers in international journals and conference proceedings. He has been involved in several national and international research projects and collaborates with several foreign research institutes. His research interests include wireless communications and networks, satellite communications and networks, edge computing, distributed learning, smart cities, and optimization techniques.

Prof. Tarchi is an Editorial Board Member of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE OPEN JOURNAL OF THE COMMUNICATION SOCIETY, and *IET Communications*. He was the Symposium Co-Chair of IEEE WCNC 2011, IEEE Globecom 2014, IEEE Globecom 2018, and IEEE ICC 2020, and the Workshop Co-Chair of IEEE ICC 2015.