# Optimizing Quality-of-Service (QoS) using Semantic Sensing and Digital-Twin in Pro-dynamic Internet of Vehicles (IoV)

Oshin Rawlley, *Student Member, IEEE,* Shashank Gupta*, *Senior Member, IEEE,* Tanisha Bhattacharyya, Shikhar Singh

2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall) | 979-8-3315-1778-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-Fall63153.2024.10757588

***Abstract*—The emerging autonomous driving industry expects real-time information to be communicated in less amount of time. Most of the extant research works on deterministic or stochastic channels, which are deemed unrealistic for pro-dynamic Internet of Vehicles (IoV) communications. Semantic communication provides a novel concept of serving high-mobility vehicles with faster vehicular communications by using digital twin (DT) technology. However, the low-latency demand, intermittent connectivity, and signal attenuation in the IoV canyon pose big challenges. To facilitate the efficient functioning of Intelligent Transport Systems (ITS) applications, we integrate DT, which is a co-simulation of software such as CARLA, SUMO, python, etc., to improve the semantic communication and quality of service (QoS) of the IoV scenario. Further, we have formulated a vehicular sensing and computation model that incorporates system cost and DT migration cost as their key metrics to evaluate the QoS of the system. We have proposed a pro-dynamic algorithm based on digital-twin deep reinforcement learning (DT-DRL) to decode the QoS maximization problem. Numerical results reveal the superiority of our method by decreasing the cost of the system and improving latency, maintaining the semantic real-time communication in IoV.**

***Index Terms*—Pro-dynamic Internet of Vehicles (IoV), Semantic sensing, Autonomous driving, age of information (AoI), low-latency, digital-twin deep reinforcement learning (DT-DRL)**

## I. Introduction

THE Internet of Vehicles (IoV), serves as the basis of the Intelligent transportation systems (ITS), interweaving vehicles, network resources, and pedestrians with cutting-edge technologies such as 6G and edge computing to facilitate a broad spectrum of applications [1]. For eg., service orchestration, traffic management, vehicle control, etc. The pro-dynamic IoV requires fast execution of intelligent vehicular applications for which the current scenario lacks certain capabilities. These capabilities encompass incompetent sensing models, low-latency communication due to high DT migration cost, etc., [2]. In order to facilitate the real-time operational feedback in the pro-dynamic IoV, the vehicles utilize a significant amount of network resources. Consequently, this burdens the sensing capability of the spectrum resources and compromises the QoS. Therefore, a paradigm shift is needed in lieu of the traditional IoV communication models [3]. Semantic communication (SC) in IoV is a novel paradigm where vehicles can sense the ambient environment using state-of-the-art edge computing devices and upload relevant data to the edge layer [4]. In this regard, SC integrates digital twin (DT) technology which captures the context-aware information and extracts semantic data i.e., demographics of the vehicles, connectivity with the edge servers (ES), etc., from the physical IoV [5]. DT technology provides the digital rendering of the physical IoV thereby, realizing model inferencing and advanced traffic analytics such as DT visualization, communication logistics, safety management, etc., as shown in *Figure* 1 [6]. DT also reduces the operational system cost of the physical management by allowing efficient virtual verification of the processes and empowers ITS with robust applications to function such as route planning, accident alert, and traffic orchestration. It overall implements a peer-to-peer feedback process and jointly caters to the issue of high latency in vehicular communication and on-demand resource orchestration [7].

In addition, the strategic deployment of ES in close proximity to the vehicles, offers supplementary computational and storage resources for handling the issue of high latency during the implementation of ITS applications. Further, different vehicular users may have varied computational preferences for quality of the channel, tolerable transmission delay, computational resources requirement, etc [8]. Presently, the sensing data comes from onboard sensors such as cameras, LIDAR, etc. Nonetheless, it is difficult to facilitate the collaborative sensing of the enormous data generated by moving vehicles. Moreover, this also results in inaccurate information collection across the entirety of the domain. This challenge is further aggravated by the incongruent sensing capabilities inherent to individual vehicles [9]. Since the ES serves the purpose of reducing system latency and transmission overhead, the DT deployment on ES facilitates fast responses with their respective physical entities [10]. Hence, the DT stationing and migration with its associated mobility costs is another fundamental problem. Therefore, it is necessary to have a dynamic sensing capability model along with a suitable DT deployment policy to meet the diverse needs of the vehicular users and enhance the QoS of the varied automotive applications [11], [12]. Therefore, we list our principal contributions as follows:

This work is supported by CHANAKYA Fellowships of IITI DRISHTI CPS Foundation under the National Mission on Interdisciplinary Cyber-Physical System (NM-ICPS) of the Department of Science and Technology, Government of India. Grant no: CF-2022-PhD-000002.

O. Rawlley, S. Gupta, T. Bhattacharyya, S. Singh are with Department of Computer Science & Information Systems, BITS Pilani, Rajasthan, India. (e-mail: p20200063@pilani.bits-pilani.ac.in, shashank.gupta@pilani.bits-pilani.ac.in, h20230197@pilani.bits-pilani.ac.in, f20221170@pilani.bits-pilani.ac.in)

979-8-3315-1778-6/24/$31.00 ©2024 IEEE

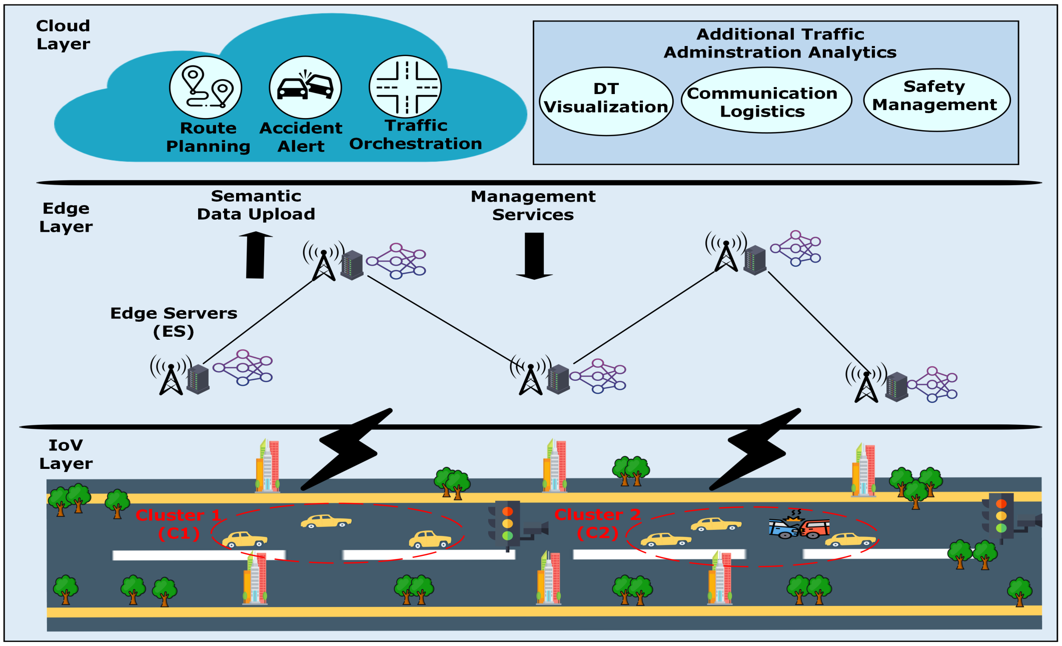


Figure 1: Proposed System Model

- We have proposed a three-tier DT-empowered pro-dynamic IoV network that allows a co-simulation of CARLA, SUMO, OSM, and python environment.
- We have developed a new paradigm of semantic communication where it utilizes DT to gauge the dynamics of the vehicle virtually. Semantic communication also focuses on the sensing capability model and DT deployment policy at the ES.
- Lastly, to solve the QoS problem, system cost and DT migration cost metrics are to solve the maximization of QoS by combining semantic sensing and Digital Twin deployment. We have developed a digital-twin deep reinforcement learning (DT-DRL) algorithm to solve the above optimization. The numerical results reveal the efficiency and convergence of DT-DRL.

## II. System Model

*Figure* 1 shows the proposed DT-empowered pro-dynamic IoV for semantic communication. It comprises a three-tier architecture having (a) a Physical IoV layer, (b) an Edge Layer, and (c) a Cloud layer. The Physical IoV layer consists of a set of vehicles $V$ $=\{v_1, v_2, v_3, ...v_k\}$ each equipped with intelligent $ES_s$ E $=\{E_1, E_2, E_3, ...E_n\}$.There are set of base stations (BSs) $B$=$\{B_1, B_2, B_3, ...B_m\}$, where $m < n$ which are capable of providing extensive communication range and advanced bandwidth to support vehicular communications with additional traffic analytics. The vehicle $v_k$ accesses $E_n$ at a time slot $t$ which is connected to the $B_m$ for uploading the status of the physical IoV along with its sensing data to update DT residing at ES. The edge layer stations the DTs where the DT of a vehicle $v_k$ is denoted as:

$$DT_k = \{Dyn_k, AE_k, RTS_k, OD_k(t + 1)\} \tag{1}$$

where, $Dyn_k$ is the dynamics of the vehicle i.e., the demographic information, speed, etc., $AE_k$ is the ambient environment with which the vehicle $v_k$ is communicating, $RTS_i$ is real-time operational status and $OD_k$ $(t + 1)$ predicts the vehicle's future state at $t + 1$ instance.

### A. Semantic Sensing model

The semantic sensing capability of a vehicle $v_k$ in terms of frequency at time instance $t$ is $SS_k(t)$ and the measure of the data generated by a single sensing process of a vehicle $v_k$ is $D_0(t)$. Then the magnitude of the vehicle's data at time instance $t$ is as follows:

$$D_i(t) = D_0(t).SS_k(t).\tau \tag{2}$$

We assume that the proposed system operates at discrete time intervals denoted as $T = \{0, 1, 2...t\}, t \in T$, where $\tau$ is the time duration. $\tau$ is infinitesimally small time duration to ensure that the policy remains unchanged. The single sensing capability is assessed on the basis of the displacement between vehicle $v_k$ and another vehicle $l$, vehicle momentum $s_{k,r,l}$, and the ambient environment $AE_k$. It can be denoted as follows:

$$Q_k = I.\frac{dist_{k,r}(t) - dist_{k,r,l}(t)}{dist_{k,r}(t)}.\frac{1}{1 + s_{k,r,l}} \tag{3}$$

where $I$ is the potency of the ambient environment on the status of sensing, $dist_{k,r}(t)$ is the maximum displacement up to which the $v_k$ can sense. The vehicle's quality of sensing is expressed as:

$$Q_k(t) = \frac{1}{ss_k(t)}.\sum_{t=1}^{ss_k(t)} Q_k \tag{4}$$

The average quality of sensing for the complete system is evaluated as:

$$Q_k^{avg} = \frac{\sum_{k=1}^{K} Q_k(t)}{K} \tag{5}$$

*B. Local Computing*

When the sub-request $D_0(t)$ is generated, it is implemented locally, making the uploading time zero. Therefore, the total implementation time taken is calculated by the CPU cycle of $v_k$ i.e., $CPU_{k,r,0}$, which is static. $r$ is the number of vehicular requests denoted as:

$$R_{k,r,0}(x_{k,r,0}) = t_{k,r,0}(x_{k,r,0}) = \frac{X_{k,r,0} D_{k,r}(t) X_{k,r}}{CPU_{k,r,0}} \tag{6}$$

where $x_{k,r,0}$ is the ratio of the sub-vehicular requests allocated to the ES itself.

*C. Uploading Model*

The rate of wireless communication from one vehicle $v_{k,r}$ to another vehicle $v_{l,r}$ or to any ES is calculated as:

$$W_{v_{k,r}}(t) = bw_{k,r,l} B_{tot} \log_2(1 + \frac{pow_{k,r,l} \left|C_{k,r,l}\right|^2 dist_{k,r,l}^{-\alpha}(t)}{\sigma_{k,r,m}^2}) \tag{7}$$

where $pow_{k,r,l}$ is the transmission power, the fading channel coefficient is $C_{k,r,l}$, $dist_{k,r,l}^{-\alpha}(t)$ is the path loss between two vehicles $k$ and $l$, $-\alpha$ is the path loss exponent exponent and $\sigma_{k,r,m}^2$ denotes the Additive White Gaussian Noise *(awgn)*. The $bw_{k,r,l} \epsilon \{0,1\}$ is the ratio of the bandwidth allocated between vehicles $v_{k,r}$ and $v_{l,r}$, whereas the total bandwidth of the IoV scenario is $B_{tot}$. As the vehicles are in continuous motion, the changing functional territories are measured by the distance between the vehicles $v_{k,r}$ and $v_{l,r}$ shown as:

$$dist_{k,r,l}(t) = \sqrt{\left|LL_{l,r} + (s_{k,r} - s_{l,r})t\right|^2}, LL_{l,r} \neq 0 \tag{8}$$

The initial positions of both the vehicles are zero and $LL_{l,r}$, given the model considers only one-way roads and $LL_{l,r} \neq 0$ because both the vehicles are not overlapping on the same position. The relative speed between two vehicles is assumed to be 60km/h. The minimum value $LL_{l,r}$ maybe a few meters.

*D. Feedback time*

After uploading and computing the tasks, the feedback is sent back to the physical IoV layer, which is denoted as:

$$t_{k,r,n}(x_{k,r,0}, bw_{k,r,l}) = \frac{x_{k,r,0} D_{k,r}}{v_{k,r} \to v_{l,r} | ES_n} = \frac{(x_{k,r,0} D_{k,r})}{bw_{k,r,l} B_{\text{tot}} \log_2 \left(1 + \frac{pow'_{k,r,l} |C'_{k,r,l}|^2 (CL_{n,r}^2 + H_{n,r}^2 + |Coord_{n,r}^2|)^{\frac{-\alpha}{2}}}{\sigma'^2_{k,r,m}}\right)} \tag{9}$$

where $CL_{n,r}^2$ is the margin from the center line of the road to the ES, $H_{n,r}^2$ is the RSU height, $Coord_{n,r}^2$ is the RSU coordinates.

*E. DT Deployment*

To define the deployment policy, we maintain a binary variable $O$ =[$o_{k,n}$], which indicates whether the $DT_k$ is deployed on ES $n$ or not. We represent it using a matrix shown as:

$$o = \begin{pmatrix} o_{1,1} & \cdots & o_{1,k} \\ \vdots & \ddots & \vdots \\ o_{K,1} & \cdots & o_{K,N} \end{pmatrix} \tag{10}$$

Owing to the constant movement of the vehicles, they frequently move away, thereby needing to adjust the position and initiate the DT migration procedure. The migration cost is determined by:

$$Mig_{cost}^k = mig_{cost}.Mem_k.dist_{n1,n2} \tag{11}$$

where, $mig_{cost}$ is the cost per unit data for per unit distance, $Mem_k$ is the memory required to host the DT, and, $dist_{n1,n2}$ is the distance between two ES. The DT migration cost for system QoS is calculated as:

$$Q_{mig_{cost}^k} = 1 - \frac{Mig_k^{cost}}{\max(Mig_k^{cost})} \tag{12}$$

*F. QoS Model*

The AoI is the newness of the data in the cloud layer expressed using $\Delta$ which tells the time duration since the previous information was refreshed. The QoS of the system is calculated using AoI which can be written as:

$$Q_{AoI} = 1 - \frac{\Delta}{th_{cloud}(t)} \tag{13}$$

where $th_{cloud}$ (t) is the threshold time after which $v_k$ will retransmit data and till which the freshness of the data is maintained. The QoS of the DT-empowered IoV system is determined mainly by sensing cost and DT migration cost. The other determinants are AoI and feedback time. AoI is expressed as:

$$\begin{gathered} Q = \gamma_1 Q_k^{avg} + \gamma_2 Q_{Mig_{cost}^k} + \gamma_3 Q_{AoI} + \gamma_4 t_{k,r,n}(x_{k,r,0}, bw_{k,r,l}) \\ \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 \in (0,1) \ and \ \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1 \end{gathered} \tag{14}$$

## III. Problem Formulation

The primary objective of the paper is to combine semantic sensing and DT deployment policy to optimize the QoS in terms of reduced system cost. It also guarantees better service and fulfills the AoI metric in DT migration. The maximization function for the QoS is expressed as follows:-

$$P : \underbrace{\max}_{SS,O} \ Q \quad st: \tag{15}$$

$$\begin{aligned} & C1 : \forall t \in T, \forall v \in V, \textstyle\sum_{n=1}^{N} o_{K,N} = 1 \\ & C2 : \forall t \in T, \forall n \in ES_n, \textstyle\sum_{k=1}^{K} o_{K,N} DTComp_k \leq ES_{Comp}^k \\ & C3 : \forall t \in T, \forall n \in ES_n, \textstyle\sum_{k=1}^{K} o_{K,N} DTMem_k \leq ES_{Mem}^k \\ & C4 : \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 \in (0,1) \\ & C5 : \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1 \\ & C6 : \forall t \in T, \forall v \in V SS_k(t) \in [0, \max(SS_k(t))] \\ & C7 : \forall t \in T, \forall v \in V, \textstyle\sum_{n=1}^{N} o_{K,N} \in \{0,1\} \end{aligned}$$

Here, the $C1$ denotes that the DT deployment is done on a unique ES at time instance $t$, $C2 - C3$ will assert that the DT will not consume all the processing capacity and the memory of the ES. $C4 - C5$ are the weights assigned to each factor contributing to the system QoS, $C_6$ tells the frequency range of the sensing model for every vehicle. $C_7$ indicates the binary decision variable.

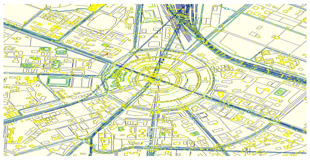

Figure 2: Open Street Map (OSM)

Figure 3: Simulation of Urban Mobility (SUMO)

## IV. Algorithm

In this section, we have proposed DT-DRL to decode the optimization problem. *Algorithm 1* implements a multi-agent reinforcement learning training loop. We consider the vehicle as the *agent* which observes the ambient state. It decides its actions based on policy to maximize its rewards. The *state* has sensing data, ES-vehicle connection, and DT deployment information of all vehicles. The *reward* is defined as the BS contribution in IoV QoS connected with the agent. We initialize the environment, Participant network parameters $A_e^a$, Arbiter network parameters $C_e^a$, and a temporary storage. During each episode = 1 to $episode_N$, our agents interact with the environment by selecting *actions* $a_{agent,step}$ based on their policies and observing rewards $r_{agent,step}$ and the next states $s_{agent,(step+1)}$. We store these transition parameters $\tau$ in the buffer. Then, for each epoch = 1 to $epochs_N$, we update $A_e^a$ and $C_e^a$ using the transition parameters $\tau$ from the buffer to improve policy and value estimations. After completing all epochs for an episode, we update the network parameters $A_e^a$ and $C_e^a$.

## V. Numerical Simulations

We construct a simulation environment as shown in *Figure* 2- 3 using SUMO, OSM, CARLA, and PyTorch framework (Python 3.10) to test our *DT-DRL* algorithm. To improve the dependability and correctness of the simulation, we have employed a trajectory dataset including data from over 500 vehicles within a 2.8-hour timeframe in a 1 sq. km area. This dataset has coordinates and timestamps of the vehicles sampled at intervals of 10 seconds. Within the simulation scenario, we have assumed the presence of 2 BS and 5 ES. This configuration enables us to mimic the semantic sensing adequately. During the Deep Reinforcement Learning (DRL) training phase, we have designed lightweight network architectures for both Actor and Critic networks. Furthermore, we have conducted experiments having hyperparameter combinations in multiple sets to identify the suitable configuration for algorithmic performance. An overview of the numerical parameters is provided in *Table I*. To prove the superiority of the proposed DT-DRL algorithm, we compare it with 3 benchmark approaches. One of them is a baseline approach and the other two are DRL schemes listed as:

Table I: Simulation Parameter Settings

| **Parameters** | **Value** |
|---|---|
| Learning rate of Actor network | 0.002 |
| Learning rate of Critic network | 0.002 |
| Discount factor | 0.99 |
| Maximum length of episodes | 400 |
| Maximum length of steps | 1000 |
| Total number of epochs | 4 |

- **Random Sensing Policy-** The vehicles sense randomly at a given time instance $t$.
- **Reinforce-** It optimizes policies based on gradients and estimates them by sampling multiple trajectories. Then, these gradients are used to improve policy gradients.
- **DDPG-** DDPG learns at every time instance $t$ by iteratively optimizing the Actor-Critic network in the pro-dynamic IoV system, thereby allowing the agent to learn the sensing model.

**Algorithm 1:** DT-DRL Algorithm

**Data:** The episodes duration $episodes_N$, the step size $steps_N$, the agent population $agents_N$, and the total number of epochs $epochs_N$.

```
Initialize the environment, Participant network
  parameters A_e^a, Arbiter network parameters C_e^a, and
  buffer.
for episodes = 1 to episodes_N do
    Reset the IoV environment
    for step = 1 to steps_N do
        for agent = 1 to agents_N do
            The agent performs an action a_{agent,step}
              based on the previous policy, generates a
              reward r_{agent,step} and observes the future
              state s_{agent,(step+1)} of environment.
            Store the transition parameters τ =
              {s_{agent,step}, a_{agent,step}, r_{agent,step}, s_{agent,(step+1)}}
              in the buffer
        end
        Update the environment state from s_{agent,step}
          to s_{agent,(step+1)}.
    end
    for epoch = 1 to epochs_N do
        for agent = 1 to agents_N do
            Retrieve a transition set τ from the buffer.
            Calculate the loss function of A_e^a based on
              τ.
            Update the parameters A_e^a.
            Calculate the loss function of C_e^a based on
              τ.
            Update the parameters C_e^a.
        end
    end
    Update A_e^a and C_e^a.
end
```

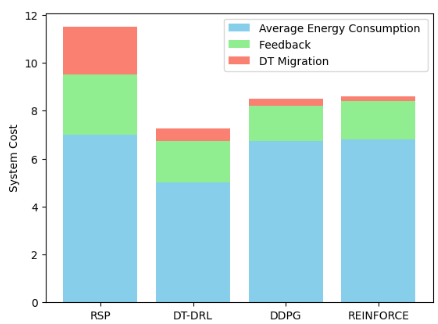


Figure 4: System Cost

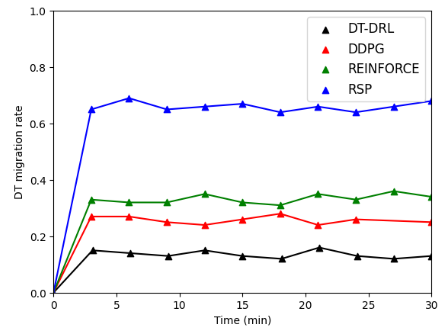


Figure 5: DT Migration Cost

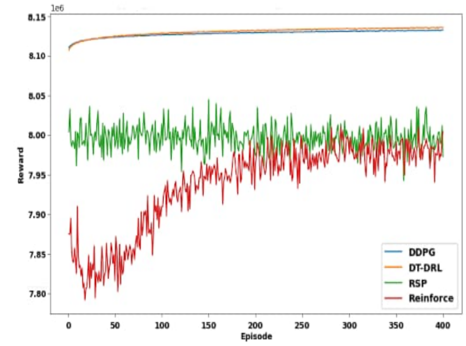


Figure 6: Rewards in training

### A. QoS Evaluation

*Figure* 4 shows the system cost of DT-DRL showing the comparison with the other three comparatives. DT-DRL reveals better performance in terms of optimization as compared with RSP, DDPG, and REINFORCE. *Figure* 5 shows the highest migration cost in the RSP approach because the vehicle tries to sense the nearest ES for DT deployment, therefore the DT relocation is initiated continuously as $v_k$ fluctuates. DT-DRL experiences the minimum cost which lowers the system cost, thereby improving the system QoS.

### B. Convergence of Algorithm 1

*Figure* 6 shows the convergence of DT-DRL *w.r.t* the average reward function. It depicts the three curves indicating the average reward after 400 episodes and compares all the benchmark approaches with the proposed DT-DRL. The initial training phase is random due to their exploration which results in erratic rewards. With the increasing number of episodes, the agent starts learning the policy better, and subsequently, the reward rises. This eventually leads to convergence, inferring that the agent has memorized the convergent policy. The DT-DRL converges at around 150 iterations. The convergence value for our algorithm is $8.135 * 1e6$, which is better than the other three schemes.

## VI. Conclusion

In this manuscript, we have proposed a pro-dynamic IoV scenario that enables semantic sensing and integrates DT technology to solve the intermittent connection in vehicular communications. For this, we chose QoS as the metric and have expressed it in many dimensions, such as semantic sensing model, feedback time, and migration cost. Subsequently, we have formulated an optimization function that maximizes the QoS in terms of the changing system cost, DT relocation cost, and rewards, as shown in the results. As the function is NP-hard, we have devised a digital-twin deep reinforcement learning (DT-DRL) algorithm, which converges to an optimal solution through many iterations. We have also compared the proposed DT-DRL algorithm with a baseline scheme and two reinforcement learning algorithms for a fair evaluation. DT-DRL has established its effectiveness and suitability in practical IoV scenario. In future, it is a potential direction to allow DT integration in ITS at the network edge for realizing semantic communication. We envisage working on a better traffic management system that simplifies interactions.

## References


[1] Sekione Reward Jeremiah, Laurence Tianruo Yang, and Jong Hyuk Park. Digital twin-assisted resource allocation framework based on edge collaboration for vehicular edge computing. *Future Generation Computer Systems*, 150:243–254, 2024.

[2] Wenjing Zhang, Yining Wang, Mingzhe Chen, Tao Luo, and Dusit Niyato. Optimization of image transmission in a cooperative semantic communication networks. *IEEE Transactions on Wireless Communications*, 2023.

[3] Xiangjie Kong, Xiaoxue Yang, Si Shen, and Guojiang Shen. Energy-delay joint optimization for task offloading in digital twin-assisted internet of vehicles. *ACM Transactions on Sensor Networks*, 2024.

[4] Jiajia Liu, Yunlong Lu, Hao Wu, and Yueyue Dai. Efficient resource allocation and semantic extraction for federated learning empowered vehicular semantic communication. In *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pages 1–5. IEEE, 2023.

[5] Junhua Wang, Kun Zhu, and Ekram Hossain. Green internet of vehicles (iov) in the 6g era: Toward sustainable vehicular communications and networking. *IEEE Transactions on Green Communications and Networking*, 6(1):391–423, 2021.

[6] Ke Zhang, Jiayu Cao, and Yan Zhang. Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks. *IEEE Transactions on Industrial Informatics*, 18(2):1405–1413, 2021.

[7] Stefan Mihai, Mahnoor Yaqoob, Dang V Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V Prasad, et al. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials*, 24(4):2255–2291, 2022.

[8] Zhenke Chen, He Li, Kaoru Ota, and Mianxiong Dong. Deep reinforcement learning for aoi aware vnf placement in multiple source systems. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 2873–2878. IEEE, 2022.

[9] Abir El Azzaoui, Sekione Reward Jeremiah, Neal N Xiong, and Jong Hyuk Park. A digital twin-based edge intelligence framework for decentralized decision in iov system. *Information Sciences*, 649:119595, 2023.

[10] Xiangjie Kong, Gaohui Duan, Mingliang Hou, Guojiang Shen, Hui Wang, Xiaoran Yan, and Mario Collotta. Deep reinforcement learning-based energy-efficient edge computing for internet of vehicles. *IEEE Transactions on Industrial Informatics*, 18(9):6308–6316, 2022.

[11] Yu Gong, Yifei Wei, Zhiyong Feng, F Richard Yu, and Yan Zhang. Resource allocation for integrated sensing and communication in digital twin enabled internet of vehicles. *IEEE Transactions on Vehicular Technology*, 72(4):4510–4524, 2022.

[12] Lun Tang, Zhangchao Cheng, Jun Dai, Hongpeng Zhang, and Qianbin Chen. Joint optimization of vehicular sensing and vehicle digital twins deployment for dt-assisted iovs. *IEEE Transactions on Vehicular Technology*, 2024.