IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 July 2023, accepted 14 August 2023, date of publication 22 August 2023, date of current version 29 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307571

RESEARCH ARTICLE

Deep Learning Based Multi-Zone AVP System Utilizing V2I Communications

ARATI KANTU KALE, MOHAMMAD SAJID SHAHRIAR, AZHARUL ISLAM, AND KYUNGHI CHANG, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Inha University, Michuhol, Incheon 22212, Republic of Korea

Corresponding author: Kyunghi Chang (khchang@inha.ac.kr)

This work was supported by the Framework of an International Cooperation Program managed by the National Research Foundation of Korea under Grant NRF-2021K1A3A1A61003229.

ABSTRACT Autonomous Valet Parking (AVP) is a technology that enables vehicles to park themselves without human intervention. It uses advanced sensing and communication systems to find a suitable parking space and to park the vehicle safely and efficiently. While various artificial intelligence (AI) based methods have demonstrated the benefits of AVP, including reducing traffic congestion, improving safety, and enhancing convenience and comfort for drivers, the issue of developing and evaluating AVP systems that can effectively handle multi-zone parking areas in real-world settings is yet to be solved. This paper presents an AVP system for three parking zones situated within a 1 km radius and utilizes a combination of existing tools and Deep Deterministic Policy Gradient (DDPG) algorithm to address the issue. DDPG algorithm controls the AVP system to allocate parking spaces efficiently in order to navigate and park vehicles autonomously. This work assumes the utilization of 5G-NR Vehicle-to-Infrastructure (V2I) communications for information exchange between vehicles and the system. It also studies the effect of communication latency on the system performance. Results of simulations show that the proposed system efficiently and safely parks vehicles in the three parking zones, achieving a reduction of 7% in waiting time compared to existing deep reinforcement learning methods. This work represents a notable advancement over current solutions and helps to advance the vision of smart cities for the future.

INDEX TERMS Autonomous valet parking, multi-zone AVP, deep deterministic policy gradient, vehicle-to-infrastructure communications.

I. INTRODUCTION

Autonomous Valet Parking (AVP) system is a technological solution that allows vehicles to park themselves in designated areas without the need for human intervention. The system uses various technologies such as sensors, cameras, and advanced algorithms to navigate the vehicle to a parking spot and park it in a safe and efficient manner. Previous research has focused on developing heuristic algorithms for solving deterministic vehicle parking tasks, but these approaches are not suitable for real-time parking space allocation in dynamic and random parking scenarios. In [1] a Reinforcement Learning (RL) based end-to-end parking algorithm is proposed for automatic parking, allowing continuous learning and optimal steering angle determination while avoiding path-tracking errors. The algorithm uses a parking slot tracking algorithm based on vision and vehicle chassis information. Authors in [2] and [3] designed and introduces the non-linear Model Predictive Control (MPC) algorithm incorporating obstacle avoidance functionalities and dynamics of vehicle based on the reference time-continuous model is given to provides the optimal solution in particular, the problem of tracking the trajectory. The goal of this study is to enhance the efficiency of parking lots by allocating parking spaces to vehicles based on the current state of the environment. This task requires a sequential decision-making process that involves perception, action, and goal elements and can be modeled using a Markov Decision Process (MDP). In their

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Mehmood.

investigation of the parking space allocation problem in an AVP system, Zhang et al. employed a Deep Reinforcement Learning (DRL) based approach described in [4]. They use their method to assign parking spaces in a small parking lot with a straightforward topological layout. The increase in automobile production has led to a strain on existing parking infrastructure, which is struggling to keep up with the rising number of vehicles. To tackle this issue, it is important to develop algorithms for autonomous valet parking systems that can manage multi-zone scenarios and cater to the growing demand for parking spaces. By incorporating machine learning algorithms and intelligent infrastructure technology into parking areas and roadsides, it is possible to enhance mobility and reduce the time taken for parking search, thereby improving overall parking efficiency. AVP solutions can be categorized into two types: Short-Range Autonomous Valet Parking (SAVP) and Long-Range Autonomous Valet Parking (LAVP) [5], [6]. LAVP allows users to drop off their vehicle and let the autonomous vehicle navigate to an available parking spot using pre-existing maps. AVP systems involve motion planning, route planning, path planning, maneuver planning, and trajectory planning [7], [8], [9]. To reduce congestion in the city center and enhance the travel experience, learning-based and DRL-based algorithms can be employed for long-range autonomous valet parking, which can optimize path planning and serving time slots. In [10] the Double-Layer Ant Colony Optimization (DL-ACO) algorithm is developed which does not require pre-training but may take longer to make decisions and consume more power. The DQN-based algorithm requires training but can make faster decisions, making it suitable for critical scenarios but (DL-ACO) algorithm may not be suitable for dynamic environments. We contribute to developing the deep learning algorithm suitable for large multi-zone dynamic environments and incorporated with intelligent infrastructures. Assuming If the Road-Side Unit (RSU) will be present in the multi-zone AVP scenario an integration of Artificial intelligence (AI) and V2X communications technologies has the potential to revolutionize the way we park our vehicles, making the process safer, more efficient, and more convenient.

The paper is structured in a systematic and comprehensive manner to offer an overview of the development of large AVP systems. Section I introduces the topic, highlighting its significance, followed by a literature review and contributions. Section II covers the brief discussion about system model. Section III provides a detailed account of the methodology employed in the study, including the problem formulation and implementation of the multi-zone AVP system algorithm and data collection methods. Section IV presents the study's findings and meticulously analyzes them. Section V discusses the experimental settings and result discussions of this study. Finally, in Section VI, the paper concludes with key implications for future research and development of large AVP systems.

A. RELATED WORKS

The emergence AVP systems offers the possibility for an Autonomous Vehicle (AV) to drop off a user at a specific location and then navigate independently to find an available parking spot. AVP systems are gaining interest due to their efficient and convenient parking solutions. DRL is a promising machine-learning technique for developing AVP systems. AVs have advanced technologies and are equipped with advanced sensors, cameras, and AI algorithms to perform driving operations autonomously and avoid obstacles. In [11] and [12] intelligent smart parking considers driver preferences and city factors to reserve parking spots, such as I-Parker, which uses Mixed Integer Linear Programming (MILP) to minimize the cost incurred by the driver. A low-cost method for vehicle detection using magnetic signals and received signal strengths is proposed in [13], which is accurate and energy-efficient, making it suitable for battery-powered wireless vehicle detectors. The study uses neural networks based on deep reinforcement learning to design and implement an autonomous prototype vehicle in [14] that can find and park in an empty parking space. The approach involves training two different artificial Neural Networks (NN) using a deep RL algorithm in a simulation environment, which is then embedded into the computing platform of the prototype car. Crowd sensing and smart parking in [15] are emerging urban services used in intelligent parking to gather real-time accessible parking information through citizens acting as parking detectors. S-Park provides real-time car park information via Onboard Unit (OBUs) and RSUs, with a Trusted Authority (TA) in [16] and [17] for authorization and a bi-linear pairing technique. Smart parking allocation & reservation system in [18] uses a cost function to assign and reserve optimal parking spaces based on current road conditions and car park information. It includes a parking geographic information system, driver request processing center, parking resource management center, and uses MILP to select the best parking space based on user-provided information. The authors proposed an algorithm in [19] for a cloud-based intelligent parking system that utilizes IoT technology to improve parking success probability and minimize user waiting time. The [20] presents a multi-objective optimization model for node placement in wireless sensor networks and suggests its relevance for smart parking and hybrid UAV/sensor networks in disaster relief. The proposed system is intended for large-scale deployment in urban areas to tackle the challenges associated with building such networks. Mukhopadhyay et al. [21] propose a smart parking lot occupancy system for smart cities that utilizes 5G communication. In this system, vehicles communicate with each other through On Board Units (OBUs) and Road Side Units (RSUs), enabling real-time updates on the availability of parking spaces. Authors in [22] focuses on integrating perception, positioning, decision-making, and maneuvering algorithms for autonomous vehicles in parking lots using a single LiDAR sensor. The system enables vehicles to search

for parking spaces based on a simplified digital map and perform accurate parking maneuvers. Challenges in infrastructure and navigation are addressed, resulting in successful tests in a real parking lot environment. A distributed Proximal Policy Optimization (PPO) algorithm with an actor-critic network is proposed in [23] for autonomous valet parking. The algorithm minimizes parking errors by guiding the ego vehicle to follow a planned path and autonomously maneuver to the target pose position. Authors in [24] focuses on the ability of AVs to park themselves. Using DRL, an agent is designed to successfully park a car in a given parking environment. Through Double Deep Reinforcement Learning, the AV achieved a 95% success rate in parking within 24 hours of training. The Python Parking Monitoring Library (PyPML) and a mobility simulation framework are introduced in [25] to address the traffic congestion and sustainable growth issues through investing in smart cities and smart mobility issues. Multiple use cases are presented to highlight the capabilities of PyPML and the need for multi-zone simulations. The framework provided a great abstraction which utilizes the Traffic Control Interface (TraCI) API.

B. CONTRIBUTIONS

Our research aims to address the limitations of previous parking management solutions, which were primarily evaluated on a small scale. Our research explores the concept of autonomous valet parking systems with the introduction of three distinct parking zones. This approach enhances the understanding and potential for efficient parking management in these systems. We strive to pave the way for the development and implementation of AI-based multi-zone AVP systems integrated with smart infrastructure. By considering communication latencies and optimizing parking resources, our work aims to improve the sustainability and efficiency of urban mobility. The contributions of this paper can be outlined as follows:

- In this study, we employ a model-free approach utilizing the DDPG algorithm to observe real-time effects on multi-zone AVP systems, we implement the DDPG scheme using PyMPL via the SUMO simulator. This approach proves to be more stable compared to existing DRL algorithms commonly used in AVP systems. By utilizing DDPG, we enable efficient and effective multi-zone parking optimizations.
- The focus of our research is on addressing the need for AI-based multi-zone AVP systems that can seamlessly integrate with smart infrastructure. By considering the integration of AVP systems with smart infrastructure, we aim to optimize urban growth sustainably while reducing waiting times for parking.
- We observe and analyze the relationship between communication latency and waiting time. The findings from these experiments offer valuable insights for reducing latencies in V2I communications. This knowledge is essential for improving the overall performance and efficiency of AVP systems, leading to reduced waiting times and enhanced user experiences.

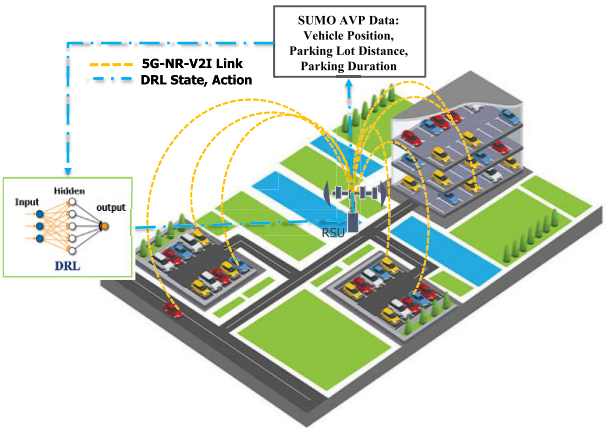


FIGURE 1. System model of DRL based multi-zone AVP system utilizing 5G-NR-V2I communications.

II. SYSTEM MODEL FOR DRL-BASED MULTI-ZONE AVP

Fig.1. illustrates a multi-zone AVP application scenario, assuming V2I communications. For intelligent multi zone AVP application systems RSU needs to be equipped with AI capabilities. In this research work we assumed that the AVP system is receiving data from the RSU utilizing V2I communications. To incorporate the RSU within the SUMO simulation, the V2I link data can supply essential details such as the position, distance, and availability status of each parking zone in the AVP scenario. In the scenario, the multi-zone AVP system communicates with a RSU installed on the roadside and provides real-time information about available parking spots, occupancy status, and other relevant data to the AVP system. Fig. 1 shows the three AVP zones situated away from each other at 700 m, 850 m, and 950 m, respectively. The parking allocation system considers the occupancy status and distance of all three parking zones to assign a suitable parking space to a vehicle. We used the SUMO simulator to create an AVP scenario. Based on the classification of autonomy levels established by the Society of Automotive Engineers (SAE), ranging from L0 (no assistance) to L5 (fully autonomous), we have proposed a solution for a multi-Zone AVP system. Our solution assumes the highest level of autonomy, L5, which signifies full autonomy in driving operations. SUMO is an open-source software that simulates traffic situations for testing and evaluating intelligent transportation systems. It models intersections, roundabouts, highways, and other complex traffic situations. In section four, we described in detail how we modeled an AVP scenario using the SUMO simulator. In this AVP system, the DDPG agent can train the multi zone AVP system to optimize its parking behaviors over time by using the DDPG algorithm; the AVP system can learn from past experiences and gradually improve its performance, resulting in a more efficient and effective parking system. We have designed the DDPG algorithm to model the

intelligent AVP system utilizing the existing parking simulation software called PyPML which also allow to use TraCI interface to model AVP scenarios in SUMO. Multi zone AVP system is designed as aims to reduce the waiting time reduction in the AVP process. In the later section we have designed and simulated the DDPG based training algorithm. The state space can be defined based on the information provided to DDPG agent from the SUMO environment, such as the number of available parking spaces, distance and occupancy in each zone, the average waiting time in each zone, and the distance between the vehicle current location and the parking zone. The action space can be defined based on the parking zone conditions i.e., distance, occupancy rates, the driver should provide the parking duration and search first. AVP system to utilize V2I communications which refers to a system where vehicles communicate with the parking infrastructure to generate parking requests and receive parking information. In this system, the vehicle sends a parking request to the infrastructure, to get the information such as the desired parking location, duration, parking distance, and occupancy rate for each parking lot. The infrastructure then evaluates the request and assigns an available parking space. The parking information is then sent back to the vehicle, which can then navigate to the assigned parking space and park itself. V2I communications is an essential aspect of the multi-zone AVP system as, it enables the vehicle and the parking infrastructure to exchange information in real time, ensuring a seamless and efficient parking experience. V2I communications serve as a vital component of the multi-zone AVP system, facilitating seamless and efficient parking operations.

III. METHODOLOGY FOR MULTI-ZONE DDPG BASED AVP SYSTEM

To evaluate our method, we select the model-free approach DDPG that can learn competitive policies for all the tasks using low-dimensional observations is a reinforcement learning algorithms commonly used for continuous control tasks, where the action space is continuous. It is a model-free, off-policy algorithm that uses deep neural networks to represent the policy and the value function. It requires only a straightforward actor-critic architecture and learning algorithm. The DPG algorithm is suitable for continuous action spaces because it uses a deterministic policy, which eliminates the need for the optimization step required by Q-learning. This makes it more computationally efficient and practical for large, unconstrained function approximators and nontrivial action spaces. DDPG, an extension of the actor-critic algorithm, involves the actor learning a deterministic policy, and the critic learning the state-value function. In comparison to other actor-critic methods, DDPG incorporates a replay buffer to save experience tuples, which are then used to train the actor and critic networks. The algorithm comprises two primary components, namely the actor and critic networks. The actor network uses the current state as input and produces the corresponding action, while the critic network utilizes the state and action to generate the expected return for that particular state-action pair. Both networks are trained using the same loss function based on the temporal difference (TD) error. The TD error is the difference between the predicted value of the current state-action pair and the actual value obtained from the reward and the next state. The loss function is a mean-squared error between the predicted and actual values. DDPG uses a mirror network for both the actor and critic networks to reduce the variance of the TD error. The mirror networks are copies of the original networks, with their weights slowly updated using a soft update rule. This helps to stabilize the training process and prevent the networks from oscillating. DDPG is a suitable algorithm for multi-zone AVP systems that consist of communication latency because it can manage partial observability and latency in receiving feedback from the environment. There can be communication latencies in autonomous valet parking systems due to several factors. These latencies can cause the autonomous vehicle to receive delayed feedback from the environment, making it challenging to control the vehicle accurately. DDPG can address this challenge by learning a policy robust to communication latencies and partial observability. The algorithm uses a replay buffer to store past experiences, which allows it to learn from delayed feedback and partial observations. Here is an example of how DDPG can be used in an autonomous valet parking system with communication delay: Suppose an autonomous vehicle is navigating a parking lot to find an available parking spot. The vehicle's sensors detect the environment and send the information to a central controller for processing. However, due to communication latencies, the central controller receives the information with a few seconds' delay. Using DDPG, the vehicle can learn a policy to manage these latencies and partial observability. The algorithm can learn from past experiences stored in the replay buffer, which allows it to adapt to communication latencies in [26] and [27] and partial observations. For example, the algorithm may learn to slow down the vehicle when it detects a potential parking spot, even if it has not received feedback from the central controller yet. This way, the vehicle can operate safely even with communication latencies. Overall, DDPG is a robust algorithm that can manage the challenges of communication latency and partial observability in autonomous valet parking systems, making it a promising approach for future autonomous vehicle technologies.

A. MARKOV DECISION PROCESS FORMULATION

The current AVP systems are limited to a single parking lot and do not have a solution for when the lot is full. Previous studies have not specified how they collect vehicle location and distance information, and they have only used DRL to optimize parking allocation in single lot scenarios. The impact of waiting time and communication latencies on AVP systems with multiple parking lots has not been thoroughly investigated. Most AVP systems are designed based on greedy approaches and only consider the single parking lot area. Although waiting time has been considered in the state

space of DRL, the effect of waiting time and communication latencies on AVP systems with multiple parking lots is yet to be explored. It is crucial to examine this aspect to understand the limitations of the current AVP systems and develop more efficient and effective solutions for parking allocation in real-world scenarios involving multiple parking lots. The proposed solution to address the limitations of existing AVP systems involves creating three separate parking zones located at different distances from each other. The system considers the occupancy and distance of all parking zones when allocating a parking space to a vehicle. We collect vehicle position and distance information using SUMO-generated data through V2I communications with a single RSU. Our DRL algorithm, specifically the DDPG algorithm, efficiently allocates parking spaces by considering occupancy, position, distance, and waiting time of vehicles in the three parking zones. In AVP systems, the average parking time (APT) is when a car travels from the drop-off location to the parking space and maneuvers into the space. The average waiting time (AWT) is when a driver waits to get their car at the pick-up point, and the average delay time (ADT) is when a car waits due to traffic conflicts while driving towards the parking space or leaving it. Our designed AVP system considers the total time a vehicle spends in the parking process, and the average waiting time is used as the performance measure to evaluate the DDPG algorithm's performance. We have designed the parking space allocation problem into a Markov decision process. The agent's actions involve allocating parking spaces in response to demand, while the environment's state includes parking space availability, vehicle location and status, and other relevant information. The policy $\pi(a_t|s_t)$ aims to maximize expected discounted rewards, including immediate and future rewards, to efficiently allocate parking spaces. The objective is to address parking congestion arising from parking space allocation for an episodic task involving the continuous arrival, parking, and departure of n vehicles. From [4] The expected discounted rewards are as follows:

$$R_t = \sum_{\tau}^{N+t-1} \gamma^{\tau-t} r_t \tag{1}$$

N corresponds to the duration of the episode, while Q^{π}(s, a) denotes the state-action value function, which represents the expected return associated with a given policy π.

$$Q^{\pi}(s,a) = E[R_t|s_t = s, a_t = a, \pi] \tag{2}$$

The fundamental aim of the MDP is to identify the optimal policy π^* that maximizes the state-action value function, as represented by:

$$Q^{\pi^*}(s,a) = max_{\pi} Q^{\pi}(s,a) \tag{3}$$

The proposed multi-zone AVP operation management architecture introduces the state space (S), action space (A), and reward function (R) in its respective section.

B. COMPONENTS OF DEEP DETERMINISTIC POLICY GRADIENT

To train the actor network, the gradient of the policy function is computed with respect to its parameters π_{μ}. This gradient is then utilized to adjust the actor-network parameters to improve the policy. However, computing the exact gradient is infeasible in continuous action spaces, where the action space is ample and uncountable. To address this, we adopt an approximation technique based on the deterministic policy gradient (DPG) algorithm. The DPG algorithm leverages the critic network's gradient as outlined in [28] to estimate the gradient of the policy network. Which is formulated as (4):

$$\nabla_{\theta^{\mu}} J \approx E[\nabla_a Q_{\theta}(s, \mu(s))|_{s=s_i} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_i} \tag{4}$$

where, $\mu(s|\theta^{\mu})$ is the actor-network that outputs the action given a state s, and Q_{θ}(s,μ(s)) the action-value function is used to estimate the expected future reward linked to a particular state-action combination. This approximation uses a gradient of the critic network for action, multiplied by the gradient of the actor-network to its parameters, evaluated at the current state s_i.The critic network is used to approximate the action-value function Q^{π}(s, a) that maps states and actions to their values. In DDPG, the critic network is trained to minimize the mean squared error (MSE) between the estimated state-action value $Q(s, a)$ and the expected state-action value y_i for each state-action pair (s, a) in the replay buffer so loss function formulated in (5):

$$Loss = 1/\mathrm{N} \sum_i (y_i - Q(s, a|\theta^Q))^2 \tag{5}$$

where N is the number of samples in the replay buffer, and θ^Q are the parameters of the critic network. The computation of the expected state-action value y_i uses the Bellman equation [29], which expresses the expected future reward as the sum of the immediate reward r_i and the discounted expected future reward given to the next state s_{i+1} and the next action a' computed by the target actor-network:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu' s_{i+1} \theta^{\mu'})|\theta^{Q'}) \tag{6}$$

$Q'(s_{i+1}, \mu'(s_{i+1}\theta^{\mu'})|\theta^{Q'})$ is the target action-value function, and γ is the discount factor, which is a copy of the critic network with frozen parameters $\theta^{Q'}$ and $\theta^{\mu'}$. The target actor-network μ' is used to compute the next action a' based on the next state s_{i+1}. The replay buffer is a dataset that stores past experiences in tuples, consisting of the state, action, reward, next state, and whether the episode is done. The DDPG algorithm enhances the stability of the training process and mitigates sample correlation by storing experiences in a buffer, from which batches can be randomly sampled for neural network training purposes. This approach is analogous to the experience replay technique utilized in the DQN algorithm, which is commonly used for tasks with discrete action spaces. Both algorithms use a replay buffer to store prior experiences, which can be randomly sampled in batches to train the neural network. This helps to break the correlation between successive samples and results in a more stable and

efficient learning process. The target networks in the dual network structure are typically represented by two separate neural networks: the target actor-network $\mu'(s|\theta^{\mu'})$ and the target critic network $Q'(s, a|\theta^{Q'})$. The actor network selects actions in the environment based on the current state, while the critic network estimates the Q-values for a given state-action pair. To update the weights of the target networks, the specific update formula used is as follows:

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{Q'} \tag{7}$$

C. PROPOSED AVP OPERATION MANAGEMENT ARCHITECTURE

Fig. 2. depicts the management architecture of the AVP system, which utilizes DRL with an Actor-Critic network called DDPG. The system architecture provides rewards based on the waiting time of a vehicle to acquire an empty parking space from any of the three available parking zones. This information is obtained from the V2I communications, which is assumed to be present in the SUMO environment. The current state S_t The Actor-Network will observe the SUMO environment', then the Actor-Network will predict an action a_t To interact back to the environment. Note that the actor-network will predict an action a_t with adding noise for exploration, according to [29].

$$a_t = \mu\,(S) + \varphi \tag{8}$$

where φ is the noise, the Ornstein-Uhlenbeck (OU) noise process is used in the implementation, as it is recommended in [26]. Moreover, the amplitude of the noise is recessing from full scale to zero to guarantee the exploration at the beginning of the learning. The environment will generate a reward r_t and will transfer to the next state S_{t+1} according to the action space. The sample (S_t, a_t, S_{t+1}, r_t) will be stored in the replay buffer. A batch of samples, N_{batch} from the replay buffer are randomly selected. Each sample consists of (S_t, a_t, S_{t+1}, r_t). For each sample, the mirror actor network will be used to predict the next action a_{t+1} from state S_t. For each sample, the mirror critic network will be used to predict the Q-value for the state-action pair S_{t+1}, a_{t+1} i.e., $Q'(S_{t+1}, a_{t+1})$. Then the Q-value for the state-action pair S_t, a_t Can be calculated according to [4] and [20] and it is denoted as y_t:

$$y_t = r_t + \gamma Q'\,(S_{t+1}, a_{t+1}) \tag{9}$$

Then (S_t, a_t, y_t) can be used for later training of the critic network. The RL algorithm uses state information to make decisions about allocating parking spaces and optimizing parking utilization in the parking lot. It should be emphasized that the state space representation outlined in [29] is specified to a particular parking lot and is being utilized to address the parking space allocation problem across three parking zones. To tackle this problem, we have expanded the representation to encompass details about multiple parking zones. This allows us to consider the distance, occupancy status, and time related information associated with each individual parking zone. Then the modified state space representation becomes:

$$\mathrm{S_{t_i}} = [\mathrm{O_{k_i}}, \mathrm{T_{k_i}}, \mathrm{t_i}] \tag{10}$$

The occupancy state variable O_{k_i} to be a concatenation of the occupancy states of all three parking zones. Let O_k Be the occupancy state vector for parking zone k, where k = 1, 2, 3 represents the parking zone ID.

- $O_{k_i} = O_{1_i}, O_{2_i}, O_{3_i}$ is the occupancy state vector for parking zone k_i.
- $T_{k_i} = T_{1_i}, T_{2_i}, T_{3_i}$ is the actual state of the parking system at a time t_i, which represents the actual conditions of the parking zones (e.g., the number of available spaces, the location of available spaces, etc.).
- t_i the time step at which the observation and state are recorded.

In the proposed system, when to normalize t_i between 0 and 1, simulations have verified that without the normalization, the value of t_i is increasing to much larger values of O_{t_i} and T_i This brings instability and a step change in t_i This will lead to high fluctuations in actor-critic network convergence; after applying such normalization, our system observed a more stabilized performance. When allocating a parking space to vehicle V_i, the chosen space is represented by action a_{t_i}, which is determined by the current state S_{t_i}. As such, the size of the action space A must match the number of available parking spaces, indicated by n:

$$a_{t_i} \in A = \{0, 1, 2, \ldots K_{K_i}, \ldots, n_{k_i}\} \tag{11}$$

the action a_{t_i} can take on any value between 0 and n_{k_i}, where n_{k_i} is the total number of available parking spaces in all three parking zones combined $(n_{1_i}, n_{2_i}, n_{3_i})$ and K_{K_i} represents an allocation of parking spaces to the vehicle in each of the three parking zones. If no parking space is allocated in a particular zone, then the corresponding index is set to 0. 0 indicates that no parking space is allocated to the vehicle. In the parking space allocation problem, the MDP utilizes a reward function to evaluate the efficacy of parking actions and determine whether the agent is adhering to the optimal allocation policy. The primary objective is to minimize the average waiting time for all vehicles. The reward at time t_i, represented as r_t, is determined by the state S_{t_i} and action a_{t_i} at that time.

$$min \frac{1}{N_{K_i}} \sum\nolimits_{i=1}^{N_{K_i}} w_i \tag{12}$$

where, N_{K_i} is the total number of parking missions from three parking zones and k = 1, 2, 3, respectively w_i is the waiting time of the owner of the vehicle V_i.

D. REWARD FORMULATION

Several metrics can be used to assess traffic congestion within a parking zone. These include the time it takes for vehicles to enter and exit the zones; the amount of time drivers must wait when retrieving their cars, and the time vehicles spend stationary during transit. In the AVP system with three

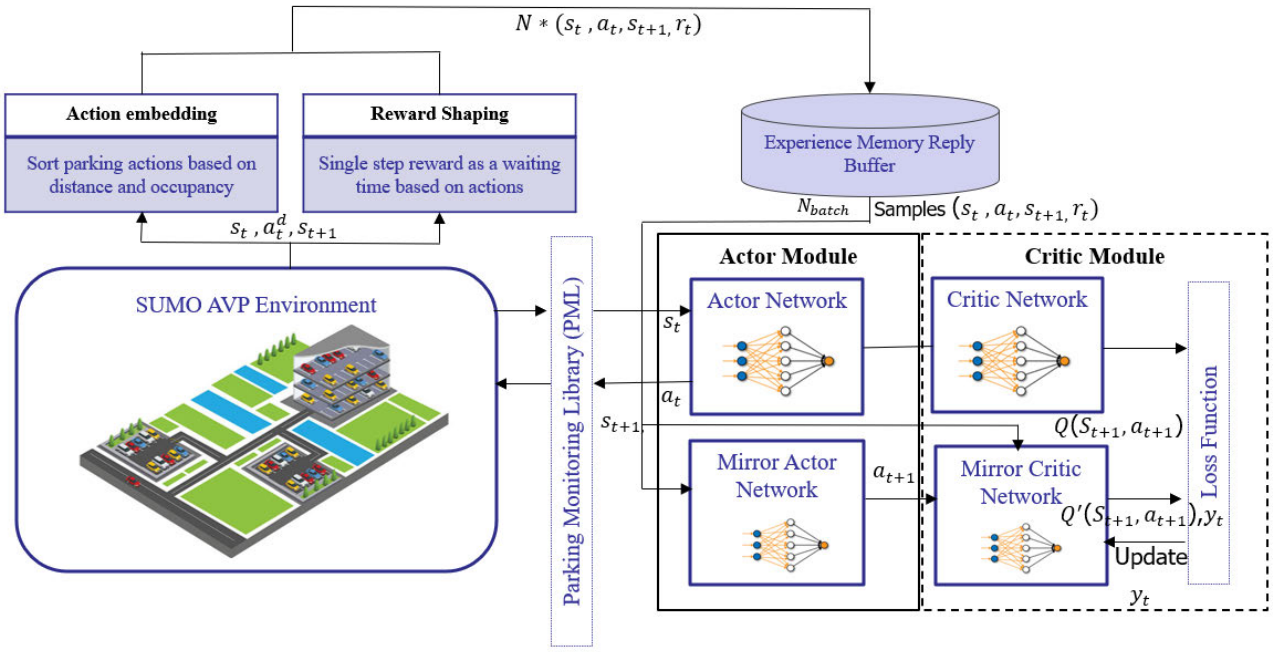


FIGURE 2. Architecture of proposed DDPG algorithm for multi-zone AVP operation management.

parking zones, we compute the average performance of N_{K_i} parking tasks to define the reward. This entails framing the problem as a periodic mission involving the allocation of N_{K_i} parking spaces across each parking zone. By implementing periodization, it is feasible to evaluate the allocation policy in a timely manner while retaining the properties of delayed feedback. We define the subsequent reward function by considering these factors and drawing upon the methodology presented in [4]. The primary metric utilized to evaluate the parking allocation system is the average time spent by N_{K_i} vehicles entering and exiting the parking zones. The time consumption comprises the total time spent driving from the highway entrance to the parking space and from the parking space to the highway exit. Considering this, we devise the reward function R_t as,

$$R_t = r_{allocated} + [r_{episode}]_{N_{K_i}}$$
$$r_{allocated} = \begin{cases} \dfrac{d_{max} - D_{K_i}}{d_{max} - d_{min}} \end{cases} \tag{13}$$

where D_{K_i} is the Where D_{K_i} is the path distance at $K_i = 1, 2, 3$, which is the path distance from the three parking zones $(d_{1_i}, d_{2_i}, d_{3_i})$ of each parking space assigned to the vehicle. V_i. d_{max} and d_{min} represent the highest and lowest values of the distance traveled along a path and $[r_{episode}]_{N_{K_i}}$. If the DRL-based allocation method finishes assigning parking spaces for n vehicles regularly, the resulting value will be 0 or less. The reward should guide a vehicle to move a short distance with less occupancy rate as proposed method. To ensure safe and valid parking, the reward system should discourage unfeasible actions that may lead to occupying already occupied parking spaces. However, allocating parking spaces solely based on distance, we have formulated the single-step reward as R_s.

$$R_s = \begin{cases} 0 - \alpha & a_{t_i} \in Emptyset_{t_i} \\ f_p - \alpha & a_{t_i} \notin Emptyset_{t_i} \end{cases}$$
$$f_p = a_n * A_{p_u} * |a_{t_i} - Emptyset_{t_i}| \tag{14}$$

In the parking allocation policy, the single-step reward R_s is defined as a function of several factors. Specifically, the factor α determines the penalty applied for incorrect parking allocations. If no parking spaces exist in any of the three parking zones, then α is set to zero. Otherwise, α is set to 0.001. Another critical factor is the negative scaling factor, a_n, which controls the magnitude of the penalty. A larger value of a_n leads to a more substantial penalty for incorrect allocations, while a smaller value results in a milder penalty. The number of available parking spaces, A_{p_u} also plays a role in determining the intensity of the penalty. When many parking spaces are available, the penalty factor is higher, as it is easier for the policy to allocate the vehicle to the correct parking zone. Conversely, when there are few available parking spaces, the penalty factor is lower, as it is more challenging for the policy to find a suitable parking space. Finally, the distance between the allocated parking space and the desired parking space also influences the penalty factor. The larger the distance, the higher the penalty factor, which encourages the policy to allocate parking spaces that are closer to the desired parking spaces. This, in turn, leads to a

more efficient use of the parking lots. These factors interact to determine the appropriate penalty applied for incorrect parking allocations. By adjusting the parameters appropriately, the parking allocation policy can be optimized for efficiency and effectiveness.

E. EMBEDDING ACTIONS FOR DISTANCE AND DURATION-BASED PARKING ALLOCATION

The action space is the set of all combinations of the available parking spaces that the autonomous vehicle can choose from. Each parking space is represented as a binary variable, where one indicates that the space is chosen and 0 indicates that it is not. We denote the action space for the parking spaces in the parking zone 1 as A1:

$$A_1 = \{a_{11}, a_{12}, a_{13}, a_{14.....}a_{1n}\} \tag{15}$$

where, $a_{1\mathrm{i}} = 1$, then i^{th} parking space is chosen. $a_{1\mathrm{i}} = 0$ if it is not chosen. Similarly, we can define the action space for the parking spaces in parking zones 2 and 3 as A_2 and A_3.

$$A_2 = \{a_{21}, a_{22}, a_{23}, a_{24.....}a_{2n}\} \tag{16}$$

$$A_3 = \{a_{31}, a_{32}, a_{33}, a_{34.....}a_{3n}\} \tag{17}$$

The occupancy factor for each parking space can be calculated using the occupancy rate of the parking zone where the parking space is located. Occupancy factor = 1 - (occupied spaces / total spaces) where occupied spaces are the number of parking spaces that are currently occupied, and total spaces are the total number of parking spaces in the zone. For example, if the parking space is in parking zone 1 and there are ten available parking spaces out of 100 total spaces, then the Occupancy Factor would be 1 - (10 / 100) = 0.9. For each parking space in the action space, calculate the distance factor for each parking zone i. In reward formation the D_{K_i} is the path distance at $K_i = 1, 2, 3$, which is the path distance from three parking zones.

$$D_{K_i} = \rho * d_{\mathrm{k_i}}^{\mathrm{h_e}} + (1-\rho)\, d_{\mathrm{k_i}}^{\mathrm{h_x}} \tag{18}$$

To consider the waiting time to park a vehicle in the empty parking space, $\rho = 0.1$. where d_{k_i} represent the distance between the highway and each parking zone $\mathrm{k_i}$ from which the parking space is assigned to a vehicle V_i. $\mathrm{h_e}$ and $\mathrm{h_x}$ represents highway distances from entry and exit. As explained in the previous section, the state space is defined based on the parking duration distribution. T_{k_i} is the true state of the parking system at the time t_i, which represents the actual conditions of the parking lots. t_i is normalized at 0 and 1 with the value of the estimated parking duration T_d of each vehicle V_i. The input data is generated using SUMO which is based on vehicles' inflow and outflow, so the parking duration for each vehicle T_i^{fac} is calculated from [29]. Estimated parking duration T_d only supports half-hour granularity, and this normalized value will impact the calculation of $Emptyset_{t_i}$. To address this, we have recalculated the T_i^{fac}.

$$T_d = [\frac{T_i^{fac}}{1800}].1800 \tag{19}$$

$$T_i^{fac} = (1-\beta)\, T_i^{P^D_{ti}} + \beta * W_T \tag{20}$$

where W_T Is the waiting time associated with the parking space from each parking zone, and it's calculated for each parking zone based on D_{K_i}; that is its distance to the highway entrance and to the exit. The default value for β is 0.1, and it can be tuned based on the condition. Then We sort all parking duration times at t_i for $\mathrm{S_{t_i}} = [\mathrm{O_{k_i}}, \mathrm{T_{k_i}}, \mathrm{t_i}]$ then we set the action space $\mathrm{A} = \{0, 1, 2, \ldots K_{K_i}, \ldots, n_{k_i}\}$ to satisfy the distance D_{K_i}.

F. CONCEPTUAL FRAMEWORK FOR DELAYED REWARD FORMATION

Taking communication latencies into account, the reward can be observed as waiting time to get empty parking space information from each parking zone. In [30], researchers propose an experience-driven approach for resource allocation in communication networks that outperforms DDPG, reducing end-to-end latency while improving total utility. We need to consider factors of distance, occupancy, and communication latency added to the waiting time to design the reward function and it is formulated in (21):

$$R_s = -((1-w)\, {}^*dist_i + w\, {}^*o_i) - d_i \tag{21}$$

where, $dist_i$ is distance between vehicle i and the central system (RSU), o_i is occupancy of parking zone i, measured as a percentage between 0 and 100, d_i is estimated communication latency for parking lot i, measured in seconds, w is a constant that determines the weight of the occupancy factor in the reward function. As before, the first two terms in the reward function calculate the distance and occupancy factors [27]. The third term subtracts the estimated communication latency for the chosen parking zone from the reward. The communication latency is subtracted because the agent wants to minimize the waiting time, and a longer latency will increase the waiting time. By combining the distance, occupancy, and communication latency factors, the agent will learn to choose the closest parking zone, with the lowest occupancy and communication delay. let us assume we set w = 0.5 as before and estimate the communication latencies as follows: 10 seconds for parking zone 1, 15 seconds for parking zone 2, and 20 seconds for parking zone 3. If parking zone 1 is chosen, the reward will be -((1 - 0.5) * 700 + 0.5 * 10) - 10 = -355. If parking zone 2 is chosen, the reward will be -((1 - 0.5) * 850 + 0.5 * 50) - 15 = -490. If parking zone 3 is chosen, the reward will be -((1 - 0.5) * 900 + 0.5 * 30) - 20 = -480. In this example, the agent will learn to choose parking zone 1 because it has the lowest combined distance, occupancy, and communication latency factor and hence the highest reward. However, the choice of the weight w and the estimation of the communication latencies will depend on the specific requirements of the AVP system and can be

adjusted as needed. The transmission latency is calculated using equation (22) and added to the reward function's waiting time. Therefore, the action space should be chosen to maximize the resulting reward value, considering both the waiting time and the transmission delay. The vehicle should choose a parking space that has the highest reward value, which can be calculated using the updated reward function. The signal transmission latency for each parking zone is calculated using Equation (22):

$$d_i = T_{hop} * ceil\left(\frac{d}{100}\right) \tag{22}$$

where d is the absolute distance between the vehicle and the RSU, and $T_{hop} = 0.005$s considering each hop is 100 meters long. In the context of Hybrid Automatic Repeat Request (HARQ) retransmissions, the number of retransmissions may also depend on the distance between the vehicle and the RSU. Specifically, if the distance is less than or equal to 100 meters, HARQ may be configured to retransmit the packet only once. If the distance is between 100 and 150 meters, HARQ may be configured to retransmit the packet up to two times, and so on. Therefore, the transmission latency may indirectly affect the number of HARQ retransmissions. If the latency is too large, the vehicles may not receive the packet correctly [31], resulting in a transmission error. This error will trigger the HARQ mechanism to initiate retransmissions. In this way, the transmission latency and HARQ retransmissions are essential to ensure reliable communication between the RSU and the vehicles. The latency affects the reliability of the initial transmission, while HARQ retransmissions provide a mechanism for correcting transmission errors and improving overall communication reliability. In the case of the AVP systems, the transmission latency is the time taken for the commands, for example:

- Parking Request: The vehicle requests the parking infrastructure to initiate the parking process.
- Parking Confirmation: The parking infrastructure confirms the availability of a parking space and sends the location and other necessary information to the vehicle.
- Parking Assistance: The parking infrastructure uses various sensors, cameras, and other technologies to guide the vehicle to the parking space.
- Parking Completion: Once the vehicle is parked in the designated space, the parking infrastructure sends a completion signal to the vehicle.
- Retrieval Request: The vehicle requests the parking infrastructure to initiate retrieval.
- Retrieval Assistance: The parking infrastructure uses various sensors, cameras, and other technologies to guide the vehicle to the exit.
- Retrieval Completion: Once the vehicle reaches the exit, the parking infrastructure sends a completion signal to the vehicle.

However, transmission over a wireless channel is prone to errors [32] due to noise, interference, and attenuation. These errors can cause the packet to be received incorrectly, resulting in a transmission error. To ensure reliable communication, 5G-NR-V2X uses HARQ, which enables the receiver to request the retransmission of lost or erroneous packets. The number of HARQ retransmissions needed depends on the quality of the wireless link, the packet size, and the distance between the transmitter and the receiver. As we discussed earlier, in the case of the parking system, the number of HARQ retransmissions depends on the distance between the vehicles and the RSU. If the transmission latency is too large, the signal travels from the RSU to the vehicle longer. This may increase the likelihood of transmission errors, triggering the HARQ mechanism to initiate retransmissions. Now, suppose that due to the distance, the packet sent by the RSU to the vehicle gets lost or corrupted during transmission. In this case, the RSU will request retransmission through the HARQ mechanism. If the number of HARQ retransmissions is set to 2, the RSU will retransmit the packet two more times [31], [33]. This means that the transmission latency for this packet will be 0.03 seconds, which is three times the original delay. As a result, the waiting time for the car will be longer than expected, which can negatively affect the performance of the parking system; detailed result discussions are explained in the result discussion section.

IV. DDPG-BASED TRAINING ALGORITHM

In reinforcement learning, selecting an action from a discrete action space often requires evaluating each action, which becomes computationally expensive when the number of actions is large. Proposed DDPG training algorithm presented in algorithm 1. This leads to an increase in the execution complexity that grows linearly with the action space size |A|. As a result, traditional value-based methods, which rely on the value function to improve the policy, become difficult to implement as the number of actions increases. To address this issue, according to [29] action embedding is one common way to choose an action: to use a policy that selects the action with the highest value according to the value function. This policy is known as a greedy policy, and it can be written as:

$$\pi_Q(s) = \text{argmax}_{a \in A} Q_{(s,a)} \tag{23}$$

In reinforcement learning, when employing a value-based approach, the value function directly impacts policy decisions. The policy's decisions are informed by the value function, which provides an estimate of the expected long-term return. Consequently, the policy selects actions based on this estimate. To promote exploration of the target policy, the agent incorporates Gaussian noise with a zero mean and gradually decreasing variance instead of using noise generated by an Ornstein-Uhlenbeck (OU) process. The OU process is utilized to generate temporally correlated exploration and enhance exploration efficiency, as described in equation (24):

$$\widehat{\mu\,(s)} = \mu\,(s) + \aleph(\mu\,(s)) \tag{24}$$

Algorithm 1 The Training Algorithm Based on DDPG

1: Randomly initialize critic network Q (s, a|θ^Q) and actor network $\mu(s|\theta^Q)$ with weight θ^μ
2: Initialize target network Q' and μ' with weights a_1, b_1
3: Initialize reply buffer R with max size S_M
4: **for** $t = 1$ to N do (N is the total number of parking missions)
5: Observe new state s_t
6: Make a prediction of the continuous action from the Actor Network
7: Calculate continuous action $\widehat{a_t} = \mu\left(s_t \mid \theta_t^\mu\right)$ from the Actor Network with addictive OU noise
8: Map from $\widehat{a_t}$ to a discrete action a_t(after considering the occupancy status of the parking zone and its neighboring zones), and reroute the vehicle according to the parking zone corresponding to the discrete action a_t
9: Map from a_t back to a continuous action $\widehat{a_t}$ (which is exact action being applied)
10: Store the transition $\left(\mathrm{s_{t-1}}, \widehat{a_{t-1}}, \mathrm{R_{t-1}}, \mathrm{s_t}\right)$ in the replay buffer //s_{t-1}, $\widehat{a_{t-1}}$, $\mathrm{R_{t-1}}$, $\mathrm{s_t}$ were already stored in a previous for-loop-iteration at $t-1$
11: Sample a minibatch of N_s transitions from the replay buffer followed by a shuffling
12: Update the Critic Network by minimizing the loss function
13: Update current policy network weight (i.e., Actor Network) by applying the policy gradient
14: Update the weights both target Actor-/Critic Networks
15: Calculate the reward $\mathbf{R_t}$
16: Store $\mathrm{s_t}$, $\widehat{a_t}$, $\mathrm{R_t}$ temporarily until the next vehicle has arrived and a new state is observed.
17: **end for**
18: **end for**

where $\aleph$ is the addictive noise correlated with the action μ(s)

$$\aleph(\mu) = \theta(\beta - \mu) + \sigma x \tag{25}$$

where x is the zero-mean unit-variance Gaussian random variable, in the experiment β is 0.5, which is relatively nearer to 1 (rather than -1), which is mapped to the parking zones near the exit to scale down the exploration when a parking zone near to the exit is allocated and to scale up the exploration when a parking zone far from the exit is allocated.

V. EXPERIMENTAL SETTINGS AND RESULT DISCUSSIONS

This section discusses the different settings of experiments and their results. Firstly, the simulation settings for the design and development of the scenario are introduced section-wise. Careful design and implementation of experiments can ensure the reliability of the results. After that, the experiments and evaluation methods of the proposed system are explained. The comparison of results to evaluate the proposed method is followed.

A. SIMULATION SETTINGS

We created a simulation platform for Deep Reinforcement Learning-based Automated Valet Parking (DRL-AVP) using SUMO and the PyMPL for three parking zone AVP system. This platform enables simulation of parking space allocation based on the distance and occupancy rate from three parking zones. We carried out a simulation experiment using SUMO to evaluate the efficiency and effectiveness of our DRL-based methodology. The experiment involved allocating parking spaces and objective is to gauge the feasibility and efficacy of our cooperative DRL approach.

1) MULTI-ZONE AVP SENARIO SETTINGS

In our implementation, we utilized the TraCI (Traffic Control Interface) provided by SUMO, which enabled our DRL-based AVP software to interact with SUMO and access information about vehicles, their locations, and other traffic-related data. We carefully considered the distances from the highway to each parking lot, which were set to 700m, 850m, and 950m, respectively, with a capacity of 10%, 30%, and 50% of occupancy rate shown in simulation Table 1. We took advantage of the advanced features available in SUMO [25], [34], [35] allowing users to define network parking zones. A parking zone is where vehicles can park and wait for a specified time. In our experiment, we calculate the vehicle parking duration based on the value of T_d. We defined parking zones using the SUMO network editor, SUMO-netedit, or by specifying them in an Extensible Markup Language (XML) file. SUMO includes several parking strategies that govern how vehicles park in a simulation, including random, closest, and discourage. It also includes features for monitoring parking occupancy, including tracking the number of vehicles parked in a specific zone and the duration of their stay. To evaluate our performance,

TABLE 1. SUMO simulation parameters.

Parameter	Value
No of parking zones	3 (namely z_1, z_2, z_3)
Parking capacity of z_1, z_2, z_3	100, 150, 300
Distance of z_1, z_2, z_3 from highway	700m, 850m, 950m
Initial occupancy rate of z_1, z_2, z_3	10%, 30%, 50%
No. of RSU	1
Flow of vehicles	2000/hr.
Average speed of vehicles	10km/h

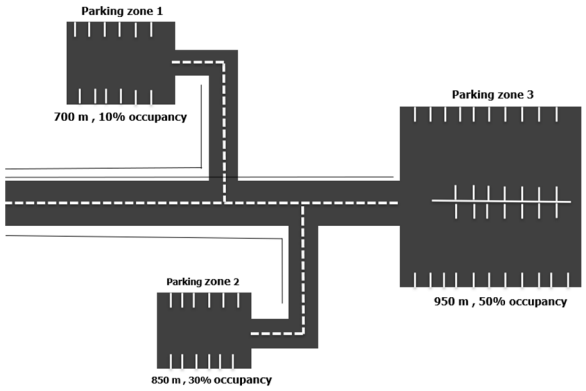


FIGURE 3. Multi-zone AVP scenario used in simulations.

we considered three methods: random, greedy, and training. The greedy and training methods were based on the DDPG algorithm, while random is a non-DRL method. We trained the greedy and training methods and evaluated the performance of DRL-based parking space allocation. According to the simulation settings parameters, the parking duration intervals were calculated using inflow and outflow from SUMO, utilizing the equation (20). Our AVP simulation system scenario is illustrated in Fig. 3. and the scenario network is built using SUMO. Specifically, parking zone 1 is designed with a capacity of 100 parking spaces, which are arranged in two rows. This zone also features two lanes, one for incoming traffic and the other for outgoing traffic. Similarly, we have constructed two additional parking zones. The second zone has a total capacity of 150 parking spaces, while the third zone has six rows of parking spaces, with each row containing 50 parking spaces. Consequently, the third zone offers a total of 300 parking spaces. Overall, the scenario includes 550 parking spaces across the three parking zones.

2) PARKING LIBRARY (PYMPL) INTEGRATION

In PyMPL, the TraCI (Online Interaction) interface is utilized to retrieve and modify the state of vehicles during the SUMO network simulation. Each parking zone is assigned a unique identifier to enable the automated valet parking space allocation algorithm, which is implemented using the DDPG algorithm. The DRL-based AVP algorithm retrieves the parking lot states and collects rewards from the Parking Monitor library. The Parking Monitor library initiates the SUMO simulation by reading the network and parking zone configuration files. Its step function retrieves the network states and modifies them accordingly. Firstly, it updates the RL learning algorithm based on the principles of DDPG. Then, it learns to allocate a parking slot for any departed vehicle in the SUMO simulation. Here, departed vehicle refers to any vehicle entering the parking network, which requests a parking zone for a certain duration. Based on the state information, the vehicle database and the parking database are updated. The AVP algorithm shuffles the parking time durations to generate random route information, which is used in each episode of the simulation. The parking durations can be rearranged randomly, enabling the addition of multiple vehicles to the simulation for larger scale AVP scenarios. This approach aims. To optimize this, the value of ρ, which controls the level of exploration, is set to 0.1 or smaller in the implementation. to reduce the AWT for drivers waiting for their cars to drive from the parking zone to the pick-up area.

3) DDPG TRAINING ALGORITHM SETTINGS

In our implementation, we have chosen a value of ρ that is equal to or smaller than 0.1 (in our specific implementation, we set ρ to 0). Our optimization target drives this decision to reduce the AWT that drivers experience when waiting for their cars to arrive from the parking zone to the pick-up area. By setting a small value for ρ, we can better achieve this objective. Our designed AVP system includes a feature that allows the generation of random route information by shuffling the parking time duration. For each training episode, a shuffled parking duration is used in one episode simulation. This functionality makes it possible to rearrange parking durations in a random order, which is advantageous for adding multiple vehicles to the simulation in multi-zone AVP scenarios. After predicting a discrete action, the next step is to apply it to the SUMO simulator. The resulting reward is collected and saved to the replay buffer, along with other relevant observations. The RL algorithm then updates its networks using batches of samples extracted from the replay buffer. In the case of departing vehicles (i.e., those that have just entered the network), our proposed system predicts a discrete action for each one. In some cases, the parking zone corresponding to the predicted action may already be occupied by other vehicles. To address this, we can search for neighboring parking zones (i.e., neighboring actions) until an unoccupied parking zone is found. Once an empty parking space is located, the Traci API is used to update the vehicle's route and parking stop information according to the predicted and corrected action. After the action is completed, the reward is collected and saved to the replay buffer. Using batches of samples from the replay buffer, the actor and critic networks in the DDPG algorithm are updated. In the methodology section, we described that T_i is the normalized value of the estimated parking durations, which is pre-calculated at the timestamp t_i. However, the argument A_{p_u} which represents the available parking spaces, should be calculated based on the current occupancies of the parking zones. To predict a continuous action, we employ the actor-network. After processing the input data, the actor-network generates a continuous action. To promote more exploration in the decision-making process, we add OU noise to the predicted action. Finally, the continuous action is mapped to a discrete action that can be executed. In the training process of the model, several hyper parameters were used to optimize its performance. These include a memory size of 100000 for experience replay, a learning rate of 0.0001 for the actor network, and a learning rate of 0.001 for the critic network. The batch size is set to 32, and the target network is

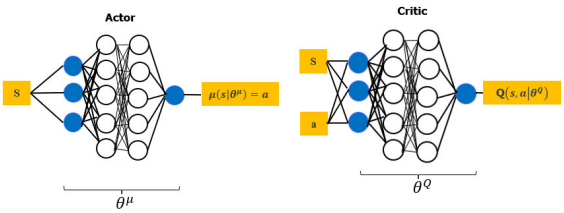


FIGURE 4. Actor network and critic network used in proposed DDPG algorithm.

updated every 50 steps. It is worth noting that the experiments were conducted based on the simulation running time, which varies depending on the number of vehicles present in the simulation. Experimental configurations for actor and critic network: The DDPG architecture utilizes fully connected neural networks for both the actor and critic networks shown in Fig. 4. The actor network takes the current state s_t as input, which corresponds to the size of the state space, and has two hidden layers with 512 processing units per layer. The activation function used in the hidden layers is ReLU. The output layer of the actor-network produces a continuous action a_t, which is activated using the hyperbolic tangent function (tanh). The critic network takes both the state s_t and the continuous action a_t as inputs, and both input and output layers are fully connected via three hidden layers, each with 512 processing units. The activation function used in the hidden layers is also ReLU. The output of the critic network is the expected reward for the given state-action pair.

B. RESULT DISCUSSIONS

To strengthen the validation and effectiveness of the allocation method based on DRL, we conducted a series of comparative experiments involving three different allocation methods which are DDPG, Greedy and Random. Here the DDPG and Greedy both methods utilize the DRL while Random method is non-DRL. When a vehicle arrives and requires a parking space, the DDPG allocation method identifies an empty space in the closest parking zone to the vehicle's current location and assigns the parking space. The greedy allocation system searches for an available parking space and assigns it to the vehicle, without considering the possibility of a more optimal or efficient allocation of parking spaces. It promptly allocates the first available parking space without further consideration. In multi zone AVP, the random allocation method is a non-DRL technique for allocating the parking spaces without considering any optimization factors. To compare the efficiency of different parking allocation methods, we kept all other parameters constant in each parking zone and measured the average time it took for vehicles to enter and exit the parking zone. The purpose of these experiments is to assess the robustness and efficiency of the DRL-based approach for large AVP systems in comparison to other commonly used methods. From Table 2, we can observe that our parking space allocation method based on

TABLE 2. Average waiting time for individual parking zone.

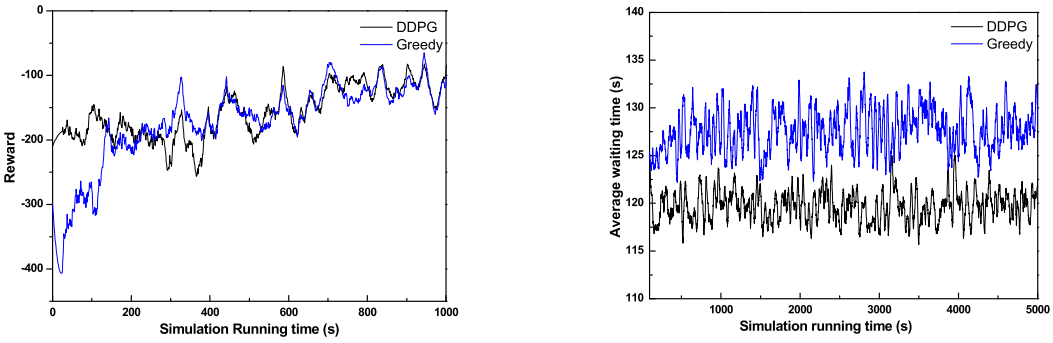
Average Waiting time	z_1 (700 m, 10% occupancy)	z_2 (850 m, 30% occupancy)	z_3 (950 m, 50% occupancy)
DDPG	125.85	127.40	128.62
Greedy	125.84	127.52	128.20
Random	134.29	134.35	134.36

TABLE 3. Average waiting time for individual parking zone considering communication latency.

Average Waiting time with delay	z_1 (700 m, 10% occupancy)	z_2 (850 m, 30% occupancy)	z_3 (950 m, 50% occupancy)
DDPG	132.92	134.35	135.21
Greedy	132.07	134.64	135.07
Random	138.07	138.35	138.14

DRL outperformed the random allocation method. In the specific case of SUMO, each episode typically corresponds to a single simulation run of a traffic scenario. Where an episode involves running the simulation for a certain duration, capturing relevant data, and updating the model parameters through the learning process.

The length of episode can vary depending on the specific requirements and objectives of the training process. In our case we have consider 2000 number of vehicles per hour. So, the length of episode depending on the number of vehicles successfully finding parking spaces, distance thresholds for parking of each vehicle, parking duration of each vehicle these factors determine when to stop the training. When all vehicles complete the parking tasks DRL model will stops learning. Fig. 5a displays the reward outcomes for both DDPG and Greedy allocation methods during the training progress, revealing a significant difference between the two approaches and shows the gradient evaluation. Fig. 5b further displays the training outcomes, revealing that the DDPG method leads to a reduction in the average waiting time, which is in seconds fluctuating between 119 and 122. In comparison, the average waiting time of the Greedy method fluctuates between 125 to 130. These results indicate that the DDPG allocation method can generate an efficient allocation policy for parking spaces from multiple parking zones



(a) (b)

FIGURE 5. (a) Average reward and (b) waiting time of DRL methods during training progress.

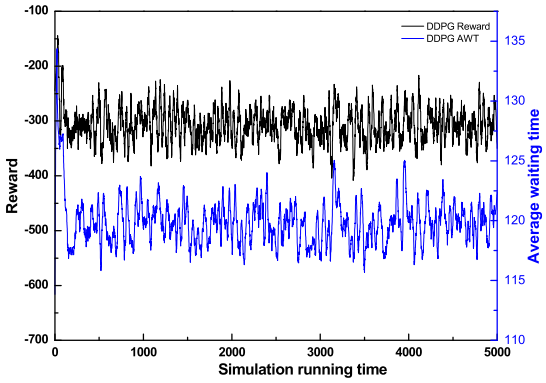


FIGURE 6. Average waiting time and reward of all parking zones during the training process of DDPG algorithm.

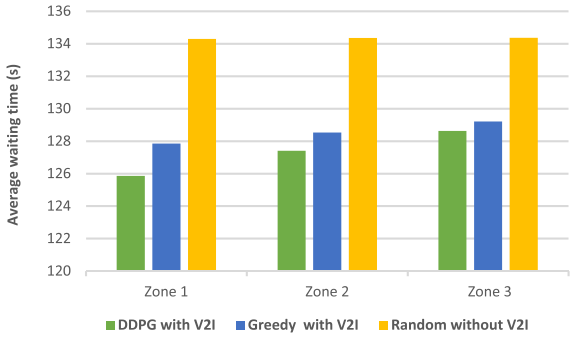
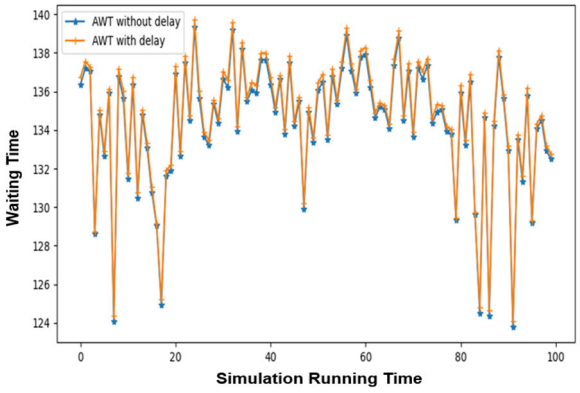


FIGURE 7. Average waiting time comparison between DRL and non-DRL method for individual parking zones.

(a) Average waiting time of each method based on distance and communication latencies.



(b) Incorporation of communication latencies in DRL training method.

FIGURE 8. Effect of Communication latencies on average waiting time.

compared with the greedy allocation method. In our experiment, parking space allocation from three parking zones is a mission planning module, and an episode refers to a single round of operation that considers global information. The planning module's input size depends on the number of available parking spaces from the three zones. To evaluate the performance of our parking allocation approach, we conducted simulations of 2000 vehicles/hr. are navigating in a multi-zone parking area and calculating the average waiting time. Fig. 6. illustrates the DDPG training results for average waiting time and reward, demonstrating the effectiveness of the DRL-based allocation method in substantially reducing waiting times across three AVP zones during the parking process. The DDPG method reveals a decrease in waiting time while in training mode for the three AVP zones, specifically when vehicles need to find the shortest distance to a park. To evaluate the efficacy of our approach, we conducted a comparison of the average waiting time for three parking zones individually shown in Fig. 7. The result shows that DRL methods which utilizes V2I communications achieves

significant improvement to reduce average waiting time compared to the random method which is non-DRL method. For instance, when the average arrival rate is 0.10 veh/s, we observed a reduction in the average waiting time from 134.36 s to 125.85 s, which points to a marked enhancement of traffic conditions and the overall efficiency of the AVP system. Our findings in Table 3 indicate that there were significant variations in time consumption among the different parking. Zones using DRL methods evaluated when adding 5ms communication latency into the waiting time. The condition of traffic in a parking zone is uncertain, which can cause a DRL agent with reliable performance in training to fail. Additionally, the agent is trained in a virtual SUMO simulation environment, which may not accurately reflect the real-world parking zone conditions, further reducing its performance. Therefore, a major challenge for DRL-based parking space allocation methods is ensuring their robustness in fluctuating large environments. Fortunately, according to [4], the difference between the real and simulated environments can be viewed as a change in environment configuration. As a result, we can evaluate the agent's generalization ability by altering the simulation environment configurations. In the methodology section, we detailed our methodology for integrating communication latency into our conceptual framework. We conducted simulations, assuming the existence of an RSU within the multi-zone AVP environment, to gauge the impact of latency on waiting time for each AVP zone. We take this step to assess the reliability of communications between various components of the AVP system. Table 3 presents our findings, illustrating the impact of latency on waiting time. Each transmission, including the retransmissions, incurs a transmission delay, which is calculated using equation (22). In Fig. 8a, shows the average waiting time at each distance for three parking zones, as the distance increases, waiting time for each vehicle increases. If the packet requires retransmission, it means that the transmission latency for that packet will be longer than expected, as it will now include the transmission latencies for the original transmission as well as the retransmissions. For example, if the original transmission latency for a packet is 0.01 seconds, and it requires two retransmissions, the total transmission latency for that packet would be 0.03 sec (0.01 sec for the original transmission + 0.01 sec for the first retransmission + 0.01 sec for the second retransmission). This latency is three times the original latency and can negatively affect the performance of the parking system. The longer transmission latency can result in a longer waiting time for a vehicle, which can negatively impact the overall performance of a parking system. This can reduce the throughput of the system, while increasing waiting time for other cars, and decrease the efficiency of the system. To address this issue, we have designed a reward function that considers both the waiting time and the transmission latency. Additionally, we have defined an action space that aims to maximize the episodic reward, which measures the performance of the reinforcement learning agent in selecting the best parking space based on the current state of the environment. To observe the impact of latency, we have introduced a 5ms communication latency into the waiting time. Fig. 8b, illustrates how this latency affects the training process of the reinforcement learning agent. Our experiment highlights the significance of considering latency in designing reward functions and training reinforcement learning agents for parking systems. It emphasizes the importance of careful optimization and evaluation to achieve efficient and reliable parking performance in real-world settings.

VI. CONCLUSION

The paper proposes an algorithm that has demonstrated impressive effectiveness in reducing waiting times in multi-zone AVP systems. Compared to the traditional random method, the DDPG algorithm achieved a remarkable 7% reduction in average waiting time, which has significant implications for the efficiency and reliability of AVP systems. This breakthrough can potentially revolutionize parking processes by reducing the need for human intervention, increasing speed, and improving the accuracy of parking operations. Combining the DDPG algorithm with the PyPML has shown tremendous potential for multi-zone AVP systems, enhancing overall system performance by reducing the waiting time using DRL methods from 134 to 127 in average, and paving the way for a more automated and efficient parking experience. Consequently, this can lead to reduced congestion, lower emissions, and more efficient use of space, contributing to a sustainable and environmentally friendly urban environment. In addition, the paper conducted experiments to assess the effect of communication latency on waiting time in AVP systems. By introducing a 5ms latency in the waiting time, the study observed the impact of communication latency on system performance. These findings provide valuable insights into enhancing communication protocols and reducing latencies in V2I communication, informing future research in AVP systems. We primarily focus on the aspects of allocating empty parking spaces and reducing waiting time in our current research. In our future work, we will actively consider incorporating factors such as pedestrians and critical situations into Multi-zone AVP systems. Overall, this research contributes to developing more efficient and reliable parking systems, with significant implications for the future of urban mobility and large multi- zone AVP systems.

REFERENCES

[1] P. Zhang, L. Xiong, Z. Yu, P. Fang, S. Yan, J. Yao, and Y. Zhou, "Reinforcement learning-based end-to-end parking for automatic parking system," *Sensors*, vol. 19, no. 18, p. 3996, Sep. 2019.

[2] P. Dini and S. Saponara, "Processor-in-the-loop validation of a gradient descent-based model predictive control for assisted driving and obstacles avoidance applications," *IEEE Access*, vol. 10, pp. 67958–67975, 2022.

[3] F. Cosimi, P. Dini, S. Giannetti, M. Petrelli, and S. Saponara, "Analysis and design of a non-linear MPC algorithm for vehicle trajectory tracking and obstacle avoidance," in *Applications in Electronics Pervading Industry, Environment and Society*. Univ. di Pisa, Jan. 2021, pp. 229–234. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-66729-0_26, doi: 10.1007/978-3-030-66729-0_26.

[4] J. Zhang, Z. Li, L. Li, Y. Li, and H. Dong, "A bi-level cooperative operation approach for AGV based automated valet parking," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103140.
[5] M. Khalid, K. Wang, N. Aslam, Y. Cao, N. Ahmad, and M. K. Khan, "From smart parking towards autonomous valet parking: A survey, challenges and future works," *J. Netw. Comput. Appl.*, vol. 175, Feb. 2021, Art. no. 102935.
[6] A. O. Hamada, F. Zahran, N. E. ElDin, M. Azab, M. Eltoweissy, and D. Gracanin, "Smartpark: A location-independent smart park and transfer system," in *Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2019, pp. 0049–0058.
[7] Y. Tazaki, H. Okuda, and T. Suzuki, "Parking trajectory planning using multiresolution state roadmaps," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 4, pp. 298–307, Dec. 2017.
[8] N. M. Nakrani and M. M. Joshi, "A human-like decision intelligence for obstacle avoidance in autonomous vehicle parking," *Int. J. Speech Technol.*, vol. 52, no. 4, pp. 3728–3747, Mar. 2022.
[9] B. Hu and S. Mishra, "Time-optimal trajectory generation for landing a quadrotor onto a moving platform," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 585–596, Apr. 2019.
[10] M. Khalid, L. Wang, K. Wang, N. Aslam, C. Pan, and Y. Cao, "Deep reinforcement learning-based long-range autonomous valet parking for smart cities," *Sustain. Cities Soc.*, vol. 89, Feb. 2023, Art. no. 104311.
[11] A. O. Kotb, Y.-C. Shen, and Y. Huang, "Smart parking guidance, monitoring and reservations: A review," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 6–16, Summer. 2017.
[12] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "IParker—A new smart car-parking system based on dynamic resource allocation and pricing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2637–2647, Sep. 2016.
[13] L. Lou, J. Zhang, Y. Xiong, and Y. Jin, "An improved roadside parking space occupancy detection method based on magnetic sensors and wireless signal strength," *Sensors*, vol. 19, no. 10, p. 2348, May 2019.
[14] A. Özeloğlu, I. G. Gürbüz, and I. San, "Deep reinforcement learning-based autonomous parking design with neural network compute accelerators," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 9, p. e6670, Apr. 2022.
[15] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017.
[16] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A new VANET-based smart parking scheme for large parking lots," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1413–1421.
[17] S. Guntuka, E. M. Shakshuki, A. Yasar, and H. Gharrad, "Vehicular data offloading by road-side units using intelligent software defined network," *Proc. Comput. Sci.*, vol. 177, pp. 151–161, Jan. 2020.
[18] Y. Geng and C. G. Cassandras, "New 'smart parking' system based on resource allocation and reservations," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1129–1139, Sep. 2013.
[19] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies," *IEEE Access*, vol. 3, pp. 1581–1591, 2015.
[20] A. Bagula, L. Castelli, and M. Zennaro, "On the design of smart parking networks in the smart cities: An optimal sensor placement model," *Sensors*, vol. 15, no. 7, pp. 15443–15467, Jun. 2015.
[21] B. Mukhopadhyay and T. Samanta, "A smart parking-lot occupancy model in 5G V2V and V2I wireless communication," in *Proc. IEEE 32nd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2021, pp. 1552–1557.
[22] F. Jiménez, M. Clavijo, and A. Cerrato, "Perception, positioning and decision-making algorithms adaptation for an autonomous valet parking system based on infrastructure reference points using one single LiDAR," *Sensors*, vol. 22, no. 3, p. 979, Jan. 2022.
[23] T. Tiong, I. Saad, K. T. K. Teo, and H. Bin Lago, "Autonomous valet parking with asynchronous advantage actor-critic proximal policy optimization," in *Proc. IEEE 12th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2022, pp. 0334–0340.
[24] M. O. Aditya, C. N. Sujatha, J. Adithya, and B. S. R. P. Kiran, "Automated valet parking using double deep Q learning," in *Proc. Int. Conf. Adv. Electron., Commun., Comput. Intell. Inf. Syst. (ICAECIS)*, 2023, pp. 259–264.
[25] L. Codecá, J. Erdmann, and J. Härri, "A SUMO-based parking management framework for large-scale smart cities simulations," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2018, pp. 1–8.
[26] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 1–7.
[27] B. Chen, M. Xu, Z. Liu, L. Li, and D. Zhao, "Delay-aware multi-agent reinforcement learning for cooperative and competitive environments," 2020, *arXiv:2005.05441*.
[28] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[29] J. Xie, Z. He, and Y. Zhu, "A DRL based cooperative approach for parking space allocation in an automated valet parking system," *Appl. Intell.*, vol. 53, pp. 5368–5387, Jun. 2022.
[30] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 1871–1879.
[31] F. Nadeem, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Nonorthogonal HARQ for URLLC: Design and analysis," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17596–17610, Dec. 2021.
[32] T. Anusha and M. Pushpalatha, "Efficient communication model for a smart parking system with multiple data consumers," *Smart Cities*, vol. 5, no. 4, pp. 1536–1553, Nov. 2022.
[33] R. E. Barone, T. Giuffrè, S. M. Siniscalchi, M. A. Morgano, and G. Tesoriere, "Architecture for parking management in smart cities," *IET Intell. Transp. Syst.*, vol. 8, no. 5, pp. 445–452, Aug. 2014.
[34] A. Hegde, R. Stahl, S. Lobo, and A. Festag, "Modeling cellular network infrastructure in SUMO," in *Proc. SUMO Conf.*, vol. 2, 2022, pp. 99–113.
[35] G.-P. Antonio and C. Maria-Dolores, "AIM5LA: A latency-aware deep reinforcement learning-based autonomous intersection management system for 5G communication networks," *Sensors*, vol. 22, no. 6, p. 2217, Mar. 2022.

ARATI KANTU KALE received the B.Eng. degree in electronics and telecommunication engineering from North Maharashtra University (NMU), Jalgaon, India, in 2017. She is currently pursuing the M.S. degree with the Department of Electrical and Computer Science Engineering (ECE), Inha University, South Korea. Since 2021, she has been a Research Assistant with the Mobile Telecommunications Research Laboratory (MTRL). Her current research interests include 5G vehicular communication and artificial intelligence.

MOHAMMAD SAJID SHAHRIAR received the B.Sc. degree from the Department of Computer Science and Engineering, International University of Business Agriculture and Technology, Bangladesh. He is currently pursuing the master's degree with the Department of Electrical and Computer Engineering, Inha University, South Korea. His research focuses on various areas, such as the Internet of Things, edge computing, artificial intelligence, and vehicular networks.

AZHARUL ISLAM received the B.Sc. degree in computer science and software engineering (CSSE) from the American International University of Bangladesh (AIUB), in 2015, and the M.Sc. degree in electronics engineering (EE) from Inha University, South Korea, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Science Engineering (ECE). Since 2019, he has been a Research Assistant with the Mobile Telecommunication Research Laboratory (MTRL). His research interests include 5G technologies, heterogeneous wireless networks, artificial intelligence, resource allocation management, and vehicular communication networks.

KYUNGHI CHANG (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Yonsei University, Seoul, South Korea, in 1985 and 1987, respectively, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 1992. From 1989 to 1990, he was with the Samsung Advanced Institute of Technology (SAIT) as a member of the Research Staff, where he has involved in digital signal processing system design. From 1992 to 2003, he was with the Electronics and Telecommunications Research Institute (ETRI) as a Principal Member of the Technical Staff, where he led the design teams involved in the WCDMA UE modem and 4G radio transmission technology (RTT). He is currently with the Electrical and Computer Engineering Department, Inha University. His research interests include radio transmission technology in 3GPP LTE and 5G NR systems, public safety, mobile ad hoc networks (especially for UAV), cellular-V2X technology, non-terrestrial networks (NTN), network intelligence for 6G, and applications of AI technologies. He was a recipient of the LG Academic Awards, in 2006; the Haedong Best Paper Awards, in 2007; the IEEE ComSoc Best Paper Awards, in 2008; the Haedong Academic Awards, in 2010; the SKT SafeNet Best Idea Awards, in 2015; the KICS Outstanding Academic Society Awards, in 2018 and 2019; the MSIT Minister's Commendation and KICS Fellow, in 2020; and the Presidential Commendation, in 2021. He is the Chairman of the 6G Forum, the Executive Committee, the Technology Committee for National Integrated Public Network, and the Expert Committee in SafeNet Forum. He has served as the Editor-in-Chief and an Executive Director for the *Journal of Korean Institute of Communications and Information Sciences* (KICS), from 2010 to 2012 and in 2013, respectively. He was a Vice President with KICS, from 2017 to 2018, and since 2021. He has also served as an Editor for ITU-R TG8/1 IMT.MOD.