Augmented Intelligence of Things for Priority-Aware Task Offloading in Vehicular Edge Computing

Xin Wang, *Member, IEEE*, Jianhui Lv, *Member, IEEE*, Adam Slowik, *Senior Member, IEEE*, Byung-Gyu Kim, *Senior Member, IEEE*, B. D. Parameshachari, *Senior Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Gang Feng, *Member, IEEE*

***Abstract*—Vehicular edge computing (VEC) systems face challenges in providing real-time intelligent transportation services due to limited computing resources at VEC servers, which lead to excessive delays or denial of services, especially for latency-critical tasks. This article proposes an augmented intelligence of things (AIoT) framework to enable priority-aware task offloading in VEC for vehicle road cooperation systems, maximizing overall system rewards under latency constraints. The framework incorporates an advanced dynamic resource management mechanism that adapts to real-time data and optimizes resource allocation using augmented intelligence models. The joint priority-aware application offloading and resource optimization problem is formulated as a constrained Markov decision process, and a deep *Q*-network (DQN)-based learning algorithm is employed to optimize the allocation of communication and computational resources based on application priorities and real-time channel/queue state information. Simulation results demonstrate that the proposed algorithm achieves significant improvements in weighted carrying capacity, high/low-priority task drop rates, and high/low-priority task queuing delays under varying overall task arrival rates, proportions of high/low-priority tasks, vehicle density, and task size compared to benchmark schemes. The proposed AIoT-enhanced DQN-based learning algorithm advances the field of VEC systems for vehicle road cooperation, offering practical advantages, such as increased efficiency, reduced latency, and improved resource utilization, ultimately enhancing user experience and enabling real-world applications in intelligent transportation systems.**

Manuscript received 29 February 2024; revised 6 April 2024, 27 April 2024, and 13 May 2024; accepted 29 May 2024. Date of publication 6 June 2024; date of current version 7 November 2024. This work was supported in part by the National Science and Technology Major Project under Grant 2022ZD0115303, and in part by the National Natural Science Foundation of China under Grant 62202247. *(Corresponding author: Jianhui Lv.)*

Xin Wang is with the School of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: dnsy_heinrich@neueet.com).

Jianhui Lv is with the Department of Network, Peng Cheng Laboratory, Shenzhen 518057, China (e-mail: lvjh@pcl.ac.cn).

Adam Slowik is with the Department of Electronics and Computer Science, Koszalin University of Technology, 75-453 Koszalin, Poland (e-mail: adam.slowik@tu.koszalin.pl).

Byung-Gyu Kim is with the Division of Artificial Intelligence Engineering, Sookmyung Women's University, Seoul 04310, Republic of Korea (e-mail: bg.kim@sookmyung.ac.kr).

B. D. Parameshachari is with the Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru 560064, India (e-mail: paramesh@nmit.ac.in).

Keqin Li is with the Department of Computer Science, The State University of New York, New Paltz, NY 12561, USA (e-mail: lik@newpaltz.edu).

Gang Feng is with the Department of Graduate, The First Affiliated Hospital of Jinzhou Medical University, Jinzhou 121012, China (e-mail: fengg@jzmu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3408157

***Index Terms*—Augmented intelligence of things (AIoT), task offloading, vehicle road cooperation, vehicular edge computing (VEC).**

I. Introduction

VEHICLE road cooperation aims to exploit advanced communication technologies and next-generation mobile Internet to realize dynamic information interaction between vehicles, humans, and infrastructure [1], [2]. Based on the collection and fusion of spatiotemporal dynamic traffic data, vehicle road cooperation systems can enable various intelligent transportation applications, such as autonomous driving, real-time traffic monitoring, and collaborative vehicle infrastructure management [3]. The key enablers underlying vehicle road cooperation include the Internet of Things (IoT) and augmented intelligence.

Combining IoT and augmented intelligence brings a new paradigm of augmented intelligence of things (AIoT) [4], [5], [6]. Augmented intelligence models combine human intelligence and machine learning algorithms to enhance decision-making processes. This article uses augmented intelligence models to optimize resource allocation and task offloading decisions in vehicle road cooperation systems. While AIoT is a paradigm that combines the power of augmented intelligence models with the vast data collected by IoT devices to enable intelligent decision-making and optimization in complex systems [7], [8], [9]. By leveraging machine learning algorithms and data analytics, AIoT can process and extract valuable insights from the heterogeneous data generated by IoT sensors, actuators, and connected devices. This integration allows for real-time monitoring, predictive maintenance, and adaptive control of IoT systems, improving efficiency, reliability, and user experience [8], [10], [11], [12].

As a key application domain of AIoT, vehicle road cooperation can benefit from AIoT in various aspects [13], [14], [15]. Massive sensors from infrastructure and vehicles can collect real-time traffic data, such as vehicle trajectories, speed, and queue length. AIoT associates surrounding vehicles and traffic participants through vehicle-to-everything (V2X) communications [16]. With user profile and preference data on smart devices, plus the global traffic analytics empowered by AIoT, customized travel recommendations and navigation can be realized to improve user experience. For autonomous vehicles, AIoT can gather surrounding traffic data from vehicles, pedestrians, and infrastructure with ultralow latency V2X connectivity [17], [18]. Safe and efficient autonomous driving can be achieved by coordinating autonomous vehicles and optimizing their paths globally.

Vehicular edge computing (VEC) has emerged as a key technology to enable real-time traffic analytics by providing cloud capabilities at the edge of wireless networks [19], [20], [21], [22], [23]. VEC systems face significant challenges in providing real-time intelligent transportation services due to the limited computing resources available at VEC servers. The cooperative vehicle edge computing system must allocate limited communication and computing resources smartly to guarantee reliable and low-latency services for mission-critical applications [24], [25], [26]. However, as application workloads from vehicles and channel conditions between vehicles and edge servers fluctuate continuously, obtaining optimal resource management policies is challenging. VEC systems face significant challenges in providing real-time intelligent transportation services due to the limited computing resources available at VEC servers.

While several studies have addressed the challenges of task offloading and resource allocation in VEC systems, there remain significant research gaps that require further investigation. The full local computing (FLC) approach relies solely on local computation at the VEC server, without considering the potential benefits of offloading tasks to nearby vehicles. The joint computation offloading and resource allocation (JCORA) [27] scheme considers both vehicle to vehicle and vehicle to roadside-unit offloading modes, but does not explicitly prioritize tasks based on their criticality. The dependency-aware task offloading and service caching (DATOSC) [28] approach formulates the problem as a mixed-integer nonlinear programming problem, which may face scalability issues in large-scale VEC systems. The belief-based task offloading algorithm (BTOA) [29] relies on a vehicle's belief about resource and channel conditions, but does not consider the dynamic nature of the vehicular environment. The location-aware and delay-minimizing task offloading (MCMTSO) [30] algorithm optimizes offloading for multiple time slots in multiple cells, but does not explicitly consider task priorities. Finally, the cross-layer cooperative offloading (CLCO) [31] algorithm focuses on rate matching between the application and MAC layers, but does not fully explore the potential of augmented intelligence models in adapting to the dynamic nature of vehicular networks. The proposed approach in this article introduces several key novelties and technical contributions that distinguish it from existing studies on task offloading in vehicle road cooperation systems.

The key contributions of applying AIoT for priority-aware intelligence in cooperative vehicle edge computing include: 1) a novel system architecture is proposed to seamlessly incorporate AIoT techniques into cooperative vehicle edge computing for dynamic workload scheduling based on heterogeneous data from end devices (e.g., vehicles and sensors), edge computing servers, and networks; 2) advanced dynamic resource management mechanism tailored for vehicle road cooperation systems is presented to maximize overall system rewards under latency constraints, with augmented intelligence models self-optimized in real time through historical operation data; and 3) an offloading strategy learning algorithm based on double deep Q-network (DQN) is proposed to optimize the allocation of communication and computational resources.

In the remainder of this article, we first introduce the system model of the cooperative vehicle edge computing system in Section II. We then formulate the joint priority-aware application offloading and resource optimization problem as a CMDP in Section III. In Section IV, we propose a deep reinforcement learning algorithm that exploits heterogeneous operation data at runtime to obtain adaptive policies for priority-differentiated application offloading and resource allocation. Performance evaluation results are presented in Section V. Finally, Section VI concludes this article.

TABLE I
SYMBOL DESCRIPTION

Symbol	Description	Symbol	Description
I	Number of vehicles	ω_H	High-priority task weight
τ	Time slot duration	ω_L	Low-priority task weight
M_i	Task computing capacity	$\overline{P}_{\max}$	Peak transmit power
$L^S_{H,\max}$	High-priority task queue length	B_i	Bandwidth
$L^S_{L,\max}$	Low-priority task queue length	$\overline{D}^{\max}_H$	High-priority task transmission delay
λ^S_H	High-priority task arrival rate	$\overline{D}^{\max}_L$	Low-priority task transmission delay
λ^S_L	Low-priority task arrival rate	θ	Task arrival rate regression model
$\lambda_{H,i}$	High-priority task arrival rate	ϕ	Vehicle computing capability prediction model
$\lambda_{L,i}$	Low-priority task arrival rate	η	Priority weight adjustment function
$f^S_{\max}$	Peak CPU frequency	δ	Threshold for Huber loss function
Z	Task input data size	γ	Discount factor
C	Task CPU cycles	ϵ	Trade-off parameter

II. SYSTEM MODEL

Table I includes the main symbols used throughout this article, along with their descriptions.

As illustrated in Fig. 1, this article considers a cooperative computing system for vehicle road infrastructure consisting of roadside edge servers equipped with limited communication and computing capacities and vehicles acting as dynamic distributed computing infrastructures. In the highly dynamic vehicle road cooperation eco-system, fluctuating user mobility patterns and wireless environments make it challenging for standalone roadside facilities to deliver reliable real-time services, calling for exploiting moving resources.

A. Task Arrival Model

In highly dynamic vehicle road cooperation systems, modeling the random task arrival process is crucial yet challenging for enabling intelligent resource management. As

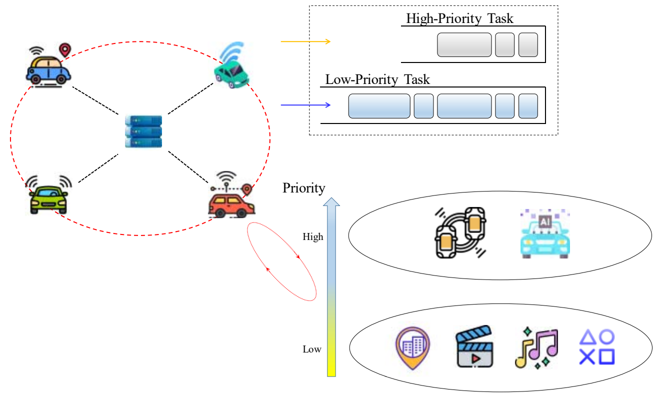

Fig. 1. System model of cooperative computing for vehicle road cooperation.

shown in Fig. 1, we consider the roadside edge server and vehicles continuously generating tasks with high or low priorities. Instead of adopting fixed Poisson processes, we establish data-driven arrival models leveraging vehicular environment data sensed by ubiquitous IoT devices in the AIoT system.

In particular, the roadside units (RSUs) can monitor surrounding traffic conditions (e.g., vehicle density and average speed). Such data reflects dynamic computational demands from vehicles, allowing more accurate arrival rate estimation through machine learning

$$\lambda_S^Z(t) = f_\theta(\text{TrafficCondition}(t)), Z \in H, L\text{TrafficCondition}(t)$$
$$= \Big[\text{VehicleFlow}(t), \text{VehicleDensity}(t)\text{AvgSpeed}(t)$$
$$\text{QueueLength}(t), \text{TravelTime}(t)]\Big] \quad (1)$$

where $f_\theta(\cdot)$ represents the data-driven arrival rate regression model parameterized by θ. The parameter θ is optimized periodically using the latest traffic data from IoT devices and historical task arrival statistics at the server. In TrafficCondition(t), VehicleFlow(t) is the number of vehicles passing a point per unit time, VehicleDensity(t) is the number of vehicles per unit length of the road, AvgSpeed(t) is the average speed of vehicles, QueueLength(t) is the number of vehicles waiting at an intersection, and TravelTime(t) is the estimated time to travel a specific route.

The parameter θ in (1) is estimated using the latest IoT device traffic data and the server's historical task arrival statistics. The real-time traffic data, such as vehicle density and average speed, is collected by roadside sensors and infrastructure, represented by TrafficCondition(t). This data is used as input to the arrival rate regression model $f_\theta(\cdot)$, which learns to predict the current task arrival rates based on the traffic conditions. The model parameter θ is updated periodically using a sliding window of the most recent traffic data and the corresponding historical task arrival rates recorded at the server. By leveraging both real-time IoT data and historical statistics, the model can adaptively capture the relationship between traffic conditions and task arrival rates, enabling more accurate predictions of $\lambda_S^Z(t)$.

Moreover, onboard IoT sensors of vehicles (e.g., advanced driver-assistance systems sensors and in-vehicle cameras) can also provide rich data about the surrounding environment as well as the state of the internal entertainment system. Such data enables estimating dynamic computational workloads and tasks generated inside vehicles using augmented intelligence

$$\lambda_{V,i}^Z(t) = f\phi(\text{InCarSensing}(t)), Z \in H, L, i \in 1, \dots, I$$
$$\text{InCarSensing}(t) = [\text{VehicleStatus}(t), \text{DriverBehavior}(t)$$
$$\text{CabinConditions}(t), \text{EntertainmentUsage}(t) \quad (2)$$

where $f_\phi(\cdot)$ denotes the arrival rate regression model for vehicles, with parameters ϕ optimized online using in-vehicle sensory data and historical task statistics. In InCarSensing(t), VehicleStatus(t) includes information about the vehicle's mechanical status, DriverBehavior(t) captures driver actions and preferences, CabinConditions(t) monitors the interior environment, and EntertainmentUsage(t) tracks the use of in-vehicle entertainment systems.

By integrating IoT sensory data closely related to vehicular computational demands, the data-driven modeling powered by AIoT can capture dynamic task arrival patterns more accurately. Thus, it enhances the awareness of fluctuating workloads across vehicles and servers, improving resource management efficiency via augmented intelligence.

B. Dynamic Queue Model

In the AIoT system for vehicle road cooperation, efficient queue management is pivotal in providing differentiated services for applications based on priorities and enabling intelligent resource optimization. As shown in Fig. 1, both the roadside edge server and onboard terminals of vehicles maintain dynamic queues for buffering unexecuted tasks.

Specifically, the roadside edge server tracks its high-priority and low-priority task queue lengths using the following AIoT-enhanced queue models:

$$L_H^S(t+1) = \min\left\{\left[L_H^S(t) - A_H^S(t) - \sum_{i=1}^{I} A_{V,i}^{OH}(t)\right]^+ + \Lambda_H^S(t), L_{\max}^S\right\} \quad (3)$$

$$L_L^S(t+1) = \min\left\{\left[L_L^S(t) - A_L^S(t) - \sum_{i=1}^{I} A_{V,i}^{OL}(t)\right]^+ + \Lambda_L^S(t), L_{\max}^S\right\} \quad (4)$$

where the numbers of high/low-priority tasks executed ($A_H^S(t)$, $A_L^S(t)$) and offloaded to vehicles ($A_{V,i}^{OH}(t)$, $A_{V,i}^{OL}(t)$) at time t are optimized dynamically based on the global queue state and channel condition information using augmented intelligence, which will be elaborated in the next section. $\Lambda_H^S(t)$ and $\Lambda_L^S(t)$ represent the high-priority and low-priority task arrival rates at the VEC server at time t, respectively. $L_{\max}^S$ denotes the maximum queue length for both high-priority and low-priority tasks at the VEC server. $A_H^S(t)$ and $A_L^S(t)$ represent the number of high-priority and low-priority tasks executed by the VEC server at time t, respectively. $A_{V,i}^{OH}(t)$ and $A_{V,i}^{OL}(t)$ denote the number of high-priority and low-priority tasks offloaded from the VEC server to vehicle i at time t, respectively. $L_H^S(t)$ and $L_L^S(t)$ represent the queue lengths for high-priority and low-priority tasks at the VEC server at time t, respectively.

Such queue models allow the server to perceive priority-differentiated application execution status precisely (e.g., queue length and waiting delay) and enhance task scheduling efficiency. Moreover, vehicle i also maintains the following queue models to track dynamic workloads:

$$L_{V,i}^{LH}(t+1) = \min\left\{\left[L_{V,i}^{LH}(t) - M_{LH,i}(t)\right]^{+} + \Lambda_{H,i}(t), L_{\max,i}^{VL}\right\} \tag{5}$$

$$L_{V,i}^{LL}(t+1) = \min\left\{\left[L_{V,i}^{LL}(t) - M_{LL,i}(t)\right]^{+} + \Lambda_{L,i}(t), L_{\max,i}^{VL}\right\} \tag{6}$$

$$L_{V,i}^{OH}(t+1) = \min\left\{L_{V,i}^{OH}(t) + A_{V,i}^{OH}(t), L_{\max,i}^{VO} - M_{OH,i}(t)\right\} \tag{7}$$

$$L_{V,i}^{OL}(t+1) = \min\left\{L_{V,i}^{OL}(t) + A_{V,i}^{OL}(t), L_{\max,i}^{VO} - M_{OL,i}(t)\right\} \tag{8}$$

where $M_{LH,i}(t)$, $M_{LL,i}(t)$, $M_{OH,i}(t)$, and $M_{OL,i}(t)$ denote the number of locally generated high-priority tasks, locally generated low-priority tasks, offloaded high-priority tasks, and offloaded low-priority tasks executed by vehicle i at time t, respectively. Equations (3) and (5) deal with high-priority tasks, while (4) and (6) focus on low-priority tasks. The distinction is crucial as it allows the system to differentiate between task priorities and allocate resources accordingly. Equations (7) and (8) further distinguish between offloaded high-priority and low-priority tasks, respectively, enabling priority-aware task offloading decisions.

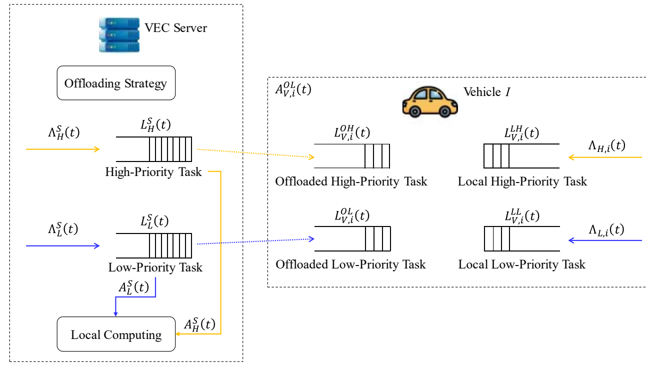

Fig. 2. Dynamic task queue model for the priority of Internet of Vehicles applications.

C. Computing Model

Efficiently allocating computational resources for vehicles and servers is vital to enable intelligent processing of dynamic workloads in-vehicle road cooperation systems [32]. The AIoT system provides a global view of real-time resource utilization and task demand levels across vehicles and edge servers. Such transparency facilitated by ubiquitous IoT interconnectivity empowers augmented intelligence to optimize dynamic computation resource allocation tailored for vehicular environments.

In particular, the roadside edge server can allocate its CPU-cycle frequency dynamically based on current workloads and application priorities

$$f^S(t) = \frac{\left(A_H^S(t) + A_L^S(t)\right)C}{\tau} \le f_{\max}^S \tag{9}$$

$$P^S(t) = \gamma\left(f^S(t)\right)^3 \tag{10}$$

where the numbers of executed high/low-priority tasks $A_H^S(t)$ and $A_L^S(t)$) are determined intelligently using augmented intelligence models by considering dynamic queues and channels. Thus, computing resource allocation and energy consumption on the server side can adapt to time-varying task demands and network conditions.

On the vehicle side, the computational capabilities are characterized by the maximum number of tasks that can be processed

$$M_i(t) = f_{\vartheta}\left(\text{CapibilityInfo}_i(t)\right) \tag{11}$$

$$M_i(t) \ge M_{LH,i}(t) + M_{LL,i}(t) + M_{OH,i}(t) + M_{OL,i}(t) \tag{12}$$

where $f_{\vartheta}(\cdot)$ represents the data-driven vehicle computing capability prediction model parameterized by ϑ. The parameters ϑ are optimized online using capability data (e.g., battery level and CPU status) streamed from in-vehicle IoT sensors and current workload statistics. The time-varying computing availability of each vehicle is estimated in real time for dynamic resource allocation. $CapibilityInfo_i(t)$ refers to the real-time capability data of the onboard computing resources of each vehicle i at time t. The dynamic task queue model for the Internet of Vehicles application priority is shown in Fig. 2.

The advanced dynamic resource management mechanism is a key component of the proposed AIoT framework, which aims to optimize the overall system rewards while satisfying latency constraints in VEC systems for vehicle road cooperation. The mechanism leverages augmented intelligence models to adaptively allocate communication and computational resources based on real-time data collected from the AIoT framework. Augmented intelligence models are trained using historical data on task demands, priority levels, resource availability, and network conditions.

The mechanism employs techniques, such as priority-based scheduling and dynamic bandwidth allocation, to optimize resource allocation under latency constraints. Priority-based scheduling ensures that latency-critical tasks are given higher priority in terms of resource allocation and execution, while dynamic bandwidth allocation adjusts the communication resources assigned to each task based on its priority and the current network conditions. The augmented intelligence models play a crucial role in enabling the mechanism to make informed decisions in real time.

In summary, the advanced dynamic resource management mechanism optimizes overall system rewards under latency constraints by leveraging augmented intelligence models to allocate resources based on real-time data adaptively. The mechanism's ability to predict future requirements, optimize allocation strategies, and adapt to dynamic conditions ensures efficient utilization of limited resources and prioritized execution of latency-critical tasks in VEC systems for vehicle road cooperation.

D. Communication Model

The highly dynamic wireless environment is one of the key characteristics of vehicle road cooperation systems, which requires efficient communication resource management adapting to fluctuating channel conditions. The AIoT system can exploit real-time channel data from ubiquitous IoT devices to empower intelligent vehicle-server communication coordination leveraging augmented intelligence.

In particular, multiple RSUs can probe wireless channel states dynamically within the coverage area to provide up-to-date channel gain information between vehicle i and the edge server [33]

$$H_i(t) = h_{\text{RSU}}(\text{ChannelProbing}(t)), i \in 1, \ldots, I \tag{13}$$

where the function $h_{\text{RSU}}(\cdot)$ aggregates channel state data from surrounding RSUs to estimate the gain. Such fine-grained channel visibility facilitates scheduling wireless resources according to network dynamics. ChannelProbing(t) refers to the real-time channel measurement data collected by the RSUs at time t, which estimates the downlink wireless channel conditions from the edge server to vehicles within the coverage area.

In the system model, a multiple-access transmission scheme lets different vehicles simultaneously offload their tasks to the edge server. One suitable scheme is orthogonal frequency division multiple access, which allocates distinct subcarriers to individual vehicles, allowing them to transmit their offloaded tasks without causing co-channel interference. However, if co-channel interference is present due to factors, such as limited orthogonal resources or imperfect resource allocation, the proposed approach can mitigate its impact through several techniques. These include power control mechanisms to adjust the transmission power of each vehicle based on the interference level, adaptive resource allocation algorithms that dynamically assign subcarriers to minimize interference and advanced receiver designs that can effectively suppress or cancel interference.

The transmission rate for offloading tasks from the edge server to vehicle i is thus given by

$$R_i\big(P_{V,i}(t), H_i(t)\big) = B_i \log_2\left(1 + \frac{P_{V,i}(t) H_i(t)}{\sigma^2}\right). \tag{14}$$

$$P_{V,i}(t) = \frac{\left(\frac{2\left(A_{V,i}^{OH}(t)+A_{V,i}^{OL}(t)\right)Z}{B_i^{\tau}} - 1\right)\sigma^2}{H_i(t)}. \tag{15}$$

Based on the channel and queue information, augmented intelligence models can determine optimal task offloading amounts $A_{V,i}^{OH}(t)$ and $A_{V,i}^{OL}(t)$, and transmission power allocation $P_{V,i}(t)$ dynamically, orchestrating vehicle-server communication efficiently.

AIoT can facilitate dynamic, intelligent wireless communication resource management tailored for vehicle road cooperation systems via data and augmented intelligence by enabling real-time channel monitoring leveraging ubiquitous connections of IoT devices.

III. Problem Modeling

Based on the system model, this section formulates the joint optimization of priority-aware application offloading and resource allocation as a CMDP problem using the framework of AIoT. The joint priority-aware application offloading and resource optimization problem is formulated as a CMDP to efficiently manage task offloading in VEC systems while considering application priorities and resource constraints. The CMDP captures the dynamic nature of the system by modeling the states, actions, rewards, and transitions, optimizing long-term system performance.

A. State and Action Spaces

The edge server can obtain real-time queue status across vehicles via cellular V2X communications empowered by AIoT

$$L_{V,i}^{ZO}(t) = h_{V2X}\big(\text{QueueReporting}_{V,i}(t)\big), Z \in LH, LL, OH, OL, \quad i \in 1, \ldots, I \tag{16}$$

where the function $h_{V2X}(\cdot)$ aggregates queue length information piggybacked in periodic basic safety messages from vehicles. *LH* is the queue length of locally buffered high-priority tasks, *LL* is the length of locally buffered low-priority tasks, *OH* is the length of offloaded high-priority tasks, and *OL* is the length of offloaded low-priority tasks. The information is periodically transmitted from vehicles to the edge server through basic safety messages or network status reporting packets. $\mathit{QueueReporting}_{V,i}(t)$ refers to the real-time queue status information of vehicle i reported at time t via V2X communications, containing queue lengths of different types of tasks buffered at the onboard unit of vehicle i.

The wireless channel gains between the edge server and vehicles can also be probed by surrounding IoT devices and transmitted using AIoT interconnectivity:

With global visibility into dynamic queues and channels enabled by AIoT, the edge server can obtain the current system state $S(t)$

$$S(t) = L_H^S(t), L_L^S(t), L_{V,i}^{LH}(t), L_{V,i}^{LL}(t), L_{V,i}^{OH}(t), L_{V,i}^{OL}(t)_{i=1}^{I}. \tag{17}$$

Based on the observed state, the edge server determines in real time the amounts of high/low-priority tasks to be executed locally or offloaded to each vehicle using augmented intelligence

$$A_U^Z(t) = f_\theta\big(S(t); \phi_U^Z(t)\big), Z \in H, L, U \in S, V \tag{18}$$

where $f_\theta(\cdot)$ represents the data-driven task scheduling policy function parameterized by θ, which are optimized periodically using historical operation data. $\phi_U^Z(t)$ denotes dynamically changing environment parameters (e.g., channel conditions and vehicle mobility patterns) that can influence task scheduling decisions.

The action space is thus given by

$$A(t) = A_H^S(t), A_L^S(t), A_{V,i}^{OH}(t), A_{V,i}^{OL}(t)_{i=1}^{I} \in \mathcal{A}. \tag{19}$$

The AIoT system can exploit cross-platform heterogeneity to dynamically orchestrate communication, computing, and storage resources tailored for vehicle road cooperation systems by enabling dynamic, transparent state monitoring and leveraging augmented intelligence for intelligent decision-making.

B. Optimization Objective

Based on the AIoT-enabled dynamic state monitoring and data-driven decision-making, the joint optimization of priority-differentiated application offloading and resource orchestration can be formulated as follows:

$$\begin{aligned}\max_{\pi} \overline{N}(\pi) &= \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_\pi[\omega_H N_H(S(t),A(t)) \\ &\qquad + \omega_L N_L(S(t),A(t))] \\ \text{s.t. } \overline{D}_H(\pi) &= \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_\pi[D_H(S(t),A(t))] \le \overline{D}_H^{\max} \\ \overline{D}_L(\pi) &= \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_\pi[D_L(S(t),A(t))] \le \overline{D}_L^{\max} \\ \overline{P}(\pi) &= \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E_\pi[P(S(t),A(t))] \le \overline{P}^{\max}. \end{aligned} \quad (20)$$

where $\overline{N}(\pi)$ represents the long-term system weighted service capability. The weights ω_H and ω_L denote differentiated priorities for high/low-priority applications, which can be dynamically adjusted over time using AIoT

$$\omega_Z(t) = f_\eta(\text{TrafficInfo}_Z(t)), Z \in H, L. \quad (21)$$

The function $f_\eta(\cdot)$ leverages augmented intelligence models to estimate real-time computational demands for high/low-priority vehicular applications based on current traffic data, and keeps tuning the weights over time. $\text{TrafficInfo}_Z(t)$ refers to the real-time traffic data related to vehicle applications with priority type Z, where $Z \in \{H, L\}$ indicates high-priority or low-priority, respectively. Tasks crucial for vehicle safety, such as collision avoidance and emergency braking, are prioritized. These tasks require immediate attention and rapid response times to ensure the safety of passengers and other road users. Other tasks, such as traffic signal optimization and route planning, are assigned lower priorities based on their impact on overall traffic efficiency and user experience. The selection of priority tasks is guided by predefined rules and policies that consider factors, such as task type, urgency, and potential consequences. Therefore, by analyzing dynamic traffic data relevant to the different application types through function $f_\eta(\cdot)$, the weight parameters $\omega_Z(t)$ can be adjusted accordingly over time to balance the service level of priorities.

Let $N_H(S(t), A(t))$, $N_L(S(t), A(t))$ denote the amounts of executed high- and low-priority tasks

$$\begin{aligned} N_H(S(t),A(t)) &= A_H^S(t) + \sum_{i=1}^{I} \min L_{V,i}^{LH}(t), M^{LH,i}(t) \\ &\quad + \sum_{i=1}^{I} \min L_{V,i}^{OH}(t) + A_{V,i}^{OH}(t), M_{OH,i}(t). \end{aligned} \quad (22)$$

$$\begin{aligned} N_L(S(t),A(t)) &= A_L^S(t) + \sum_{i=1}^{I} \min L_{V,i}^{LL}(t), M_{LL,i}(t) \\ &\quad + \sum_{i=1}^{I} \min L_{V,i}^{OL}(t) + A_{V,i}^{OL}(t), M_{OL,i}(t). \end{aligned} \quad (23)$$

The constraints represent QoS guarantees for high/low-priority applications in terms of transmission delays and power consumption

$$D_H(S(t),A(t)) = \frac{L_H^S(t) + \max_i L_{V,i}^{LH}(t)}{\lambda_H^S + \sum i = 1^I \lambda_{H,i}}. \quad (24)$$

$$D_L(S(t),A(t)) = \frac{L_L^S(t) + \max_i L_{V,i}^{LL}(t)}{\lambda_L^S + \sum i = 1^I \lambda_{L,i}}. \quad (25)$$

$$P(S(t),A(t)) = \sum_{i=1}^{I} \frac{\left(\frac{2\left(A_{V,i}^{OH}(t)+A_{V,i}^{OL}(t)\right)Z}{B_i^\tau} - 1\right)\sigma^2}{H_i(t)}. \quad (26)$$

By leveraging dynamic queues/channels monitoring and cross-platform data sharing empowered by AIoT to optimize priority-differentiated application offloading and resource orchestration online using augmented intelligence, the quality of service for mission-critical applications can be guaranteed based on their priorities.

IV. Algorithm Design and Implementation

To obtain optimal policies for the AIoT-enhanced joint optimization problem formulated in the last section, we propose a deep reinforcement learning algorithm leveraging heterogeneous data across vehicles, edge servers, and networks.

A. Lagrangian Method

The constrained MDP formulation allows the mathematical articulating of an ideal dynamic resource orchestration scheme across communication, computing, and content caching unifying systems for vehicle road coordination applications by distinguishing prioritized tiers based on criticality levels. However, deriving optimal policy is analytically intractable due to state space explosion resulting from numerous intertwined workflows, including generation, offloading, communication, execution, energy, and content libraries [34]. Fortuitously, the AIoT architecture's meshed sensing topology generates fine-grained snapshots capturing multidimensional runtime factors. The proposed double DQN methodology achieves model-free optimal coordinated control for edge computing in vehicle road systems at the system level. By leveraging the AIoT architecture's meshed sensing topology, the algorithm generates fine-grained snapshots capturing multidimensional runtime factors, enabling the algorithm to make informed offloading decisions and adaptively allocate resources in real time, effectively addressing the challenges of the dynamic and heterogeneous vehicle road system ecosystem. The neural networks used in the proposed algorithms consist of an input layer, two hidden layers, and an output layer. The input layer contains state information, such as queue lengths, channel conditions, and task characteristics. The hidden layers use the ReLU activation function and have 64 and 32 neurons, respectively. The output layer has a linear activation function and outputs the Q-values for each possible offloading decision.

The adaptive learning-based task offloading algorithm is designed to dynamically learn the optimal offloading policy in the presence of uncertainty and variability in the vehicular network environment. The multiarmed bandit theory inspires the algorithm, which is a framework for decision-making under uncertainty. In task offloading, each possible offloading decision (e.g., offloading to a specific vehicle or executing locally) is considered an arm of the bandit. The algorithm learns each arm's expected rewards (e.g., reduced latency and improved throughput) through repeated interactions with the environment.

Specifically, the ubiquitous connections empowered by IoT allow dynamic sharing queue backlogs and channel conditions across the server and vehicles

$$Q_Z(t) = h_{V2X}(\text{QueueReporting}(t)), Z \in H, L \tag{27}$$

$$H_i(t) = h_{\text{RSU}}(\text{ChannelProbing}(t)), i \in 1, \ldots, I \tag{28}$$

where $Q_Z(t)$ represents the queue length of tasks with priority Z at time t, where $Z \in H, L$.

Based on the globally visible state, the lagrangian function is

$$\begin{aligned} &C(\overline{\omega}, S(t), A(t)) \\ &= -\omega_H N_H(S(t), A(t)) - \omega_L N_L(S(t), A(t)) + \omega_1 D_H(Q_H(t)) \\ &-\overline{D}_H^{\max}\omega_2 D_L(Q_L(t)) - \overline{D}_L^{\max} \\ &+\omega_3\left(\sum_{i=1}^{I}\frac{\left(\frac{2\left(A_{V,i}^{OH}(t)+A_{V,i}^{OL}(t)\right)Z}{B_i^{\tau}} - 1\right)\sigma^2}{H_i(t)} - \overline{P}^{\max}\right). \end{aligned} \tag{29}$$

The primal problem becomes

$$G(\pi, \overline{\omega}) = \max_{\omega \geq 0} \min_{\pi} \sum_{S(t)} d_\pi(S(t))\pi(S(t), A(t))C(\overline{\omega}, S(t), A(t)) \tag{30}$$

where $d_\pi(S(t))$ is the state distribution under policy π. The global information facilitated by AIoT transforms the partial derivative of a Lagrangian function to the Bellman equation. It enables adopting reinforcement learning to find an optimal policy without requiring system dynamics. Next, we introduce a DQN algorithm that exploits heterogeneous operation data across vehicles, edge servers, and networks empowered by AIoT to obtain optimal policy online.

B. DQN-Based Offloading Policy Learning

As the AIoT architecture provides transparent and real-time state information availability encompassing dynamic workloads, communication dynamics, and resource configurations, the partial derivative of the Lagrangian function equals the Bellman equation. The connection motivates the adoption of modern reinforcement learning techniques for online optimization without requiring an a priori knowledge of system dynamics. We propose an innovative independent double DQN algorithm that interacts with the target vehicle-edge-cloud cooperative computing environment powered by AIoT. By using two separate Q-networks (online and target), Double DQN provides more stable and accurate Q-value estimates, leading to improved learning performance. Double DQN is also more sample-efficient and computationally lighter than actor-critic methods, making it more suitable for resource-constrained VEC environments. Experience replay in Double DQN further enhances its data efficiency and stability, enabling faster convergence and better adaptability to dynamic network conditions.

The proposed DQN-based offloading algorithm introduces several technical novelties, including a state space representation, that incorporates task priority information and vehicle mobility patterns, a reward function that balances task completion time and energy consumption while prioritizing critical tasks, and a training process that employs experience replay and target network stabilization techniques to improve convergence and adaptability to dynamic vehicular environments. First, it integrates the concept of priority-aware task offloading, allowing the system to dynamically adapt its offloading decisions based on the criticality and time sensitivity of tasks. Second, the DQN algorithm is enhanced with a multiobjective reward function that considers both the priority of tasks and the overall system performance. Furthermore, the proposed approach incorporates a novel state representation that encapsulates the dynamic nature of the vehicular environment, including task characteristics, resource availability, and network conditions. The rich state representation allows the DQN agent to make informed decisions based on a comprehensive understanding of the current system state.

Cooperative computing in-vehicle road cooperation systems involve the collaboration between vehicles and edge servers to execute tasks efficiently and optimize resource utilization. In cooperative computing, tasks are offloaded from vehicles to edge servers and other vehicles based on task priority, resource availability, and network conditions. High-priority tasks are given precedence in offloading decisions to ensure their timely completion. Resource allocation mechanisms are employed to distribute computing and communication resources among vehicles and edge servers, considering the dynamic nature of the vehicular environment. These mechanisms maximize resource utilization while ensuring fair distribution among participating entities. Data sharing and synchronization are crucial in enabling collaborative decision-making and optimization.

The rationale behind using a DQN-based learning algorithm for optimizing resource allocation in the proposed AIoT-enhanced framework lies in its ability to effectively handle large state spaces and learn optimal policies through environmental interaction. Through continuous interaction with the environment and experience replay, DQN can learn and adapt its resource allocation policies in real time, improving system performance and efficiency.

The key of DQN is the action-value function $Q(s, a; \theta)$ modeled by deep neural networks with weights θ [35]. It represents the expected long-term accumulated rewards after taking a at state s. $Q(s, a; \theta)$ is optimized by minimizing the loss function using the tuples of state transition and immediate reward (s, a, r, s') stored in replay memory D

$$L(\theta) = \mathbb{E}\big(s, a, r, s'\big) \sim D\big[L_\delta(y, Q(s, a; \theta))\big] \tag{31}$$

Algorithm 1 AIoT-Enhanced DQN-Based Offloading Policy Learning

Input: Vehicle number I; maximum episode L; threshold ϵ.
Output: Optimal policy π^*.

```
Initialize replay memory D; action-value function Q(s, a; θ) with random weights θ; target network Q(s, a; θ⁻) = θ⁻ = θ.
For (episode = 1 : L)
    Server obtain state s = Q_H, Q_L, H_i^I using AIoT.
    For (t = 1 : T)
        if probability 1 − ε then
            a ← arg max_a Q(s, a; θ).
        else
            a ← random action.
        end-if
        Obtain task arrivals Λ_H(t), Λ_L(t)_i^I from vehicles via AIoT.
        Take action a, reach s', store (s, a, r, s') in D.
        Sample minibatch from D; Update θ by optimizing L(θ).
        Update target network weights θ⁻.
    end-for
end-for
π* = arg ! max_π Q(s, π(s); θ).
```

where the Huber loss is

$$L_\delta(y, u) = \begin{cases} \frac{1}{2}(y-u)^2, & \text{if } |y-u| \le \delta \\ \delta|y-u| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \tag{32}$$

where y and u are variables used in the definition of the Huber loss function $L_\delta(y, u)$. The Huber loss function is used in the DQN-based offloading algorithm because it combines the advantages of both mean squared error (MSE) and mean absolute error (MAE) loss functions. For minor errors, Huber loss behaves like MSE, which is more sensitive to outliers and helps the model converge faster. It behaves like MAE for large errors, which is more robust to outliers and prevents the model from being heavily influenced by extreme values. This balance between sensitivity and robustness makes Huber loss particularly suitable for the proposed approach, as it can handle the dynamic and noisy nature of the vehicular network environment while ensuring stable and efficient learning of the offloading policy.

The target value $y = r + \gamma \max_{a'} (s', a'; \theta^-)$ where θ^- is weights of target network. The detailed steps of the proposed DQN-based algorithm are described in Algorithm 1.

Algorithm 1 presents the AIoT-enhanced DQN-based offloading policy learning process. The algorithm starts by initializing the replay memory, Q-networks, and target networks. The VEC server observes the current state at each episode and selects an action based on the epsilon-greedy strategy. The selected action is executed, and the next state and reward are observed. The experience tuple is stored in the replay memory for training. The Q-network is updated using a mini-batch sampled from the replay memory, and the target network is periodically synchronized with the Q-network.

The offloading strategy learning algorithm based on DQN adaptively optimizes task offloading decisions by leveraging real-time channel/queue state information and application priorities. The algorithm learns from the collected data by continuously interacting with the VEC environment and updating its offloading policies based on the observed rewards. At each time step, the algorithm inputs the system's current state, including channel conditions, queue lengths, and application priorities. It selects an action (i.e., offloading decision) based on its current policy. The resulting reward (e.g., reduced latency and increased throughput) is then used to update the Q-values and adjust the policy using the DQN's neural network.

The proposed AIoT-enhanced DQN-based learning algorithm optimizes the allocation of communication and computational resources in VEC systems for vehicle road cooperation. The algorithm follows a step-by-step process, which includes data collection, preprocessing, model training, and optimization. The AIoT framework enables real-time data collection from various sources, such as vehicles, VEC servers, and network infrastructure. Once the DQN-based learning algorithm is trained, it can optimize the allocation of communication and computational resources in real time. The algorithm takes the current state of the VEC system, including application demands, task priorities, channel conditions, and resource availability, as input. It generates optimal offloading decisions and resource allocation strategies.

V. Simulation and Results Analysis

This article proposes AIoT-enhanced DQN-based learning for priority-aware vehicular task offloading in cooperative edge systems (AIoT-DQN-PAVET). In this section, we evaluate the performance of the proposed AIoT-DQN-PAVET algorithm by comparing it with six benchmark schemes, including FLC, JCORA [27], DATOSC [28], BTOA [29], MCMTSO [30], and CLCO [31].

A. Simulation Setup

We consider an edge computing network consisting of 1 VEC server co-located with a RSU and I vehicles. The key simulation parameters are set as per Table II. Moreover, the learning rate was set to 0.001, which allows for a gradual and stable update of the Q-network weights. The discount factor was set to 0.99, giving more importance to future rewards and promoting long-term optimization. The exploration-exploitation tradeoff was managed using an epsilon-greedy strategy, with epsilon decreasing from 1 to 0.1 throughout training. These hyperparameters were chosen based on empirical tuning and their impact on the convergence speed and stability of the learning process. The simulation experiments were conducted on a computer equipped with an Intel Core i7-14700KF processor (20 cores, 28 threads), 32-GB DDR4-3200-MHz RAM, and an NVIDIA GeForce RTX 3080 GPU (10-GB VRAM). The experiments were run on the GPU to accelerate the training and inference processes of the DQN algorithm.

The performance of the proposed algorithm is evaluated using several key metrics, including weighted carrying

TABLE II
PARAMETER SETTINGS

Parameter	Value
I	[1,4]
τ	1 ms
M_i	2 tasks
$L_{S_H}^{\max}$	8 tasks
$L_{S_L}^{\max}$	8 tasks
λ_{S_H}	[1,8] tasks/τ
λ_{S_L}	[1,8] tasks/τ
$\lambda_{H,i}$	0.5 tasks/τ
$\lambda_{L,i}$	0.5 tasks/τ
$f_S^{\max}$	8×10^8 cycles/s
Z	2×10^3 bits
C	2×10^5 cycles
ω_H	100
ω_L	1
$\bar{P}^{\max}$	2 W
B_i	180 kHz
$\bar{D}_H^{\max}$	1 ms
$\bar{D}_L^{\max}$	10 ms

capacity, high/low-priority task drop rate, high/low-priority task queuing delay, and overall system rewards. These metrics were chosen to comprehensively assess the algorithm's ability to efficiently handle task offloading and resource allocation in VEC systems while prioritizing latency-critical tasks. The weighted carrying capacity measures the system's overall throughput, considering the different tasks' priorities. The high/low-priority task drop rates indicate the algorithm's effectiveness in completing latency-critical tasks. The high/low-priority task queuing delays reflect the algorithm's ability to minimize latency for critical tasks. System rewards quantify the algorithm's performance using a weighted combination of the above metrics. These metrics were measured through extensive simulations under various scenarios, including different task arrival rates, priority distributions, and network conditions, to demonstrate the algorithm's robustness and adaptability.

B. Results and Analysis

1) Varying Overall Task Arrival Rates: In the first set of simulations, we vary the overall task arrival rates on the VEC server from 1 to 8 tasks/τ to evaluate the performance as the total workload imposed on the system increases. The key results are summarized in Fig. 3.

AIoT-DQN-PAVET attains a prominently lower drop rate for high- and low-priority tasks under increasing arrival rates by exploiting priority differentiation and adapting task offloading and resource orchestration using augmented intelligence.

2) Varying Proportions of High/Low-Priority Tasks: In the second set of simulations, we evaluate a mismatched case between high- and low-priority task arrival rates, where the high-priority traffic is lighter than the low-priority one. We consider two cases: 1) the high-priority traffic arrival rate is lower than the low-priority traffic arrival rate (1:2 ratio) and

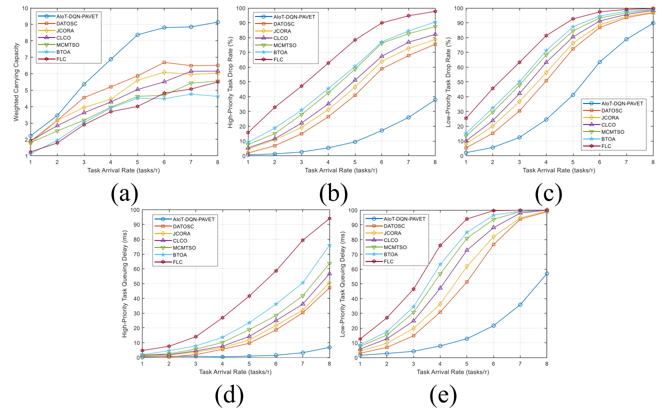

Fig. 3. Performance under varying overall task arrival rates. (a) Weighted carrying capacity. (b) High-priority task drop rate. (c) Low-priority task drop rate. (d) Highpriority task queuing delay. (e) Low-priority task queuing delay.

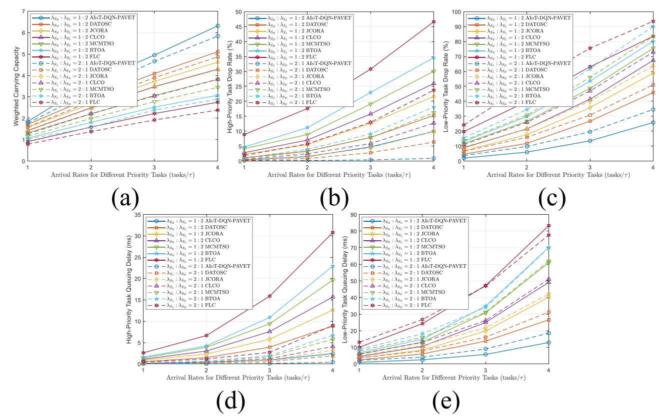

Fig. 4. Performance under varying proportions of high/low-priority tasks. (a) Weighted carrying capacity. (b) High-priority task drop rate. (c) Low-priority task drop rate. (d) High-priority task queuing delay. (e) Low-priority task queuing delay.

2) the high-priority traffic arrival rate is higher than the low-priority traffic arrival rate (2:1 ratio). The rationale behind having a higher high-priority traffic arrival rate in the second case is to assess the algorithm's ability to prioritize critical tasks and ensure timely completion, even when they constitute a larger portion of the overall traffic. The results are exhibited in Fig. 4.

The weighted carrying capacity attained by AIoT-DQN-PAVET in both 1:2 and 2:1 cases in Fig. 4 notably surpasses other solutions. This validates its capability to exploit spare resources to handle more low-priority tasks when the high-priority traffic load is lighter.

3) Impact of Vehicle Density: In the third set of simulations, we vary the number of vehicles from 1 to 4 to evaluate the performance under different vehicle densities, and the results are exhibited in Fig. 5.

The proposed AIoT-DQN-PAVET algorithm realizes superior weighted carrying capacity over other solutions when available vehicles are limited, as shown in Fig. 5. This verifies its ability to exploit potential resources on sparse vehicles fully enabled by AIoT. It also accomplishes a significantly lower drop rate for high- and low-priority tasks by judiciously offloading tasks between the edge server and vehicles based

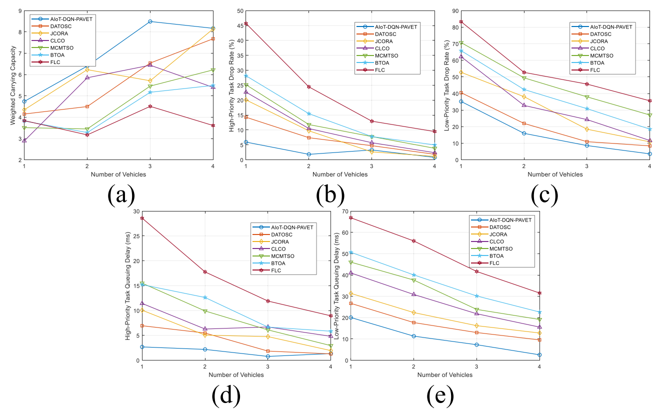

Fig. 5. Performance under varying vehicle density. (a) Weighted carrying capacity. (b) Highpriority task drop rate. (c) Low-priority task drop rate. (d) High-priority task queuing delay. (e) Low-priority task queuing delay.

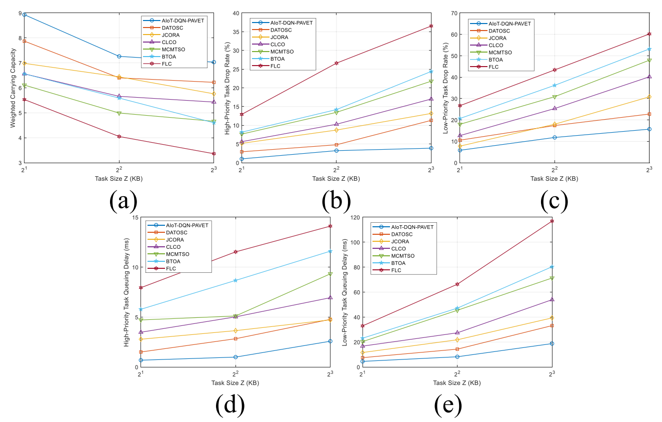

Fig. 6. Performance under varying task size. (a) Weighted carrying capacity. (b) Highpriority task drop rate. (c) Low-priority task drop rate. (d) High-priority task queuing delay. (e) Low-priority task queuing delay.

on application priorities. However, AIoT-DQN-PAVET maintains its superior performance by efficiently leveraging the additional resources through intelligent offloading decisions. JCORA and DATOSC perform slightly better than the other benchmarks as they consider the heterogeneity of vehicles and tasks in their offloading decisions. BTOA and MCMTSO, while showing improvements over FLC, do not fully exploit the potential of cooperative computing, resulting in suboptimal performance. Despite considering cross-layer optimization, CLCO needs to adapt effectively to the dynamic nature of the vehicular environment.

4) Impact of Task Size: We further evaluate the impact of the input data size of each task by increasing Z from 2 to 4 and 8 kB while keeping other settings unchanged, where Z represents the task input data size, measured in bits. The results in Fig. 6 and the key trends plotted manifest that the gain of AIoT-DQN-PAVET in weighted carrying capability over other solutions becomes larger as tasks become more resource-demanding with higher Z in Fig. 6.

The proposed algorithm demonstrates robust performance and maintains its advantages over benchmark schemes under varying conditions. When task arrival rates increase, the algorithm efficiently manages the higher workload by adaptively allocating resources and prioritizing latency-critical tasks, ensuring minimal drop rates and queuing delays. As the proportion of high/low-priority tasks changes, the algorithm adjusts its offloading decisions to ensure that high-priority tasks receive the necessary resources while maintaining good performance for low-priority tasks. With increasing vehicle density, the algorithm effectively leverages the additional resources to improve overall system performance. Finally, as task sizes vary, the algorithm adapts its offloading strategies to optimize resource utilization and minimize latency.

5) Overhead and Latency Analysis: To clarify the overhead required for the proposed AIoT-DQN-PAVET approach and its impact on latency, we consider the additional communication and computation costs introduced by the AIoT framework and the DQN-based offloading algorithm. The main overhead sources include data collection and transmission, state information exchange, and computation overhead. The simulation results in Table III reflect the impact of overhead on latency for the proposed AIoT-DQN-PAVET algorithm and the benchmark schemes.

TABLE III
COMPARISON OF AVERAGE LATENCY FOR DIFFERENT ALGORITHMS

Algorithm	Average Latency (ms)
AIoT-DQN-PAVET	85.6
FLC	150.3
JCORA [27]	120.8
DATOSC [28]	110.5
BTOA [29]	105.2
MCMTSO [30]	98.7
CLCO [31]	92.4

Table III shows that the proposed AIoT-DQN-PAVET algorithm achieves the lowest average latency among all the compared schemes, even when considering the overhead. This can be attributed to the efficient offloading decisions made by the DQN-based algorithm, which considers the real-time system state and adapts to the dynamic network conditions. DATOSC, BTOA, MCMTSO, and CLCO perform better than FLC and JCORA but still have higher latencies than AIoT-DQN-PAVET. This is because these schemes do not fully leverage the capabilities of AIoT and DQN for real-time data-driven decision-making and optimization.

Further, we conducted additional simulations and analysis to provide a more detailed breakdown of the overhead and its impact on latency. The main sources of overhead in the proposed AIoT-DQN-PAVET algorithm include the following.

1) *Channel Estimation:* The AIoT framework requires periodic channel probing and information exchange between vehicles and edge servers to maintain up-to-date channel state information, introducing additional communication overhead.
2) *Data Collection and Transmission:* The AIoT framework relies on the collection and transmission of real-time data from various sources, such as vehicles, edge servers, and network infrastructure.

TABLE IV
AVERAGE LATENCY UNDER DIFFERENT SCENARIOS

Scenario	Average Latency (ms)
Ideal (no overhead)	78.2
With channel estimation	81.4
With data collection	83.1
With state exchange	84.5
With computation overhead	85.6
Full AIoT-DQN-PAVET	87.3

3) *State Information Exchange:* To enable informed decision-making by the DQN-based offloading algorithm, the current system state information needs to be exchanged between vehicles and edge servers regularly, including queue lengths, channel conditions, and task characteristics, contributing to the overhead.
4) *Computation Overhead:* The DQN-based offloading algorithm introduces computational overhead due to the training and inference of the neural networks.

To separate the latency caused by the overhead from the protocol-related latency, we have conducted simulations to measure the average latency under different scenarios. The results are presented in Table IV.

Table IV shows that each source of overhead contributes to an increase in the average latency compared to the ideal scenario without any overhead. Channel estimation adds around 3.2 ms; data collection contributes 4.9 ms; state information exchange adds 6.3 ms, and computation overhead accounts for 7.4 ms. The full AIoT-DQN-PAVET approach, which includes all the overhead sources, has an average latency of 87.3 ms, which is still lower than the benchmark schemes. It is important to note that while the overhead does introduce additional latency, the AIoT-DQN-PAVET approach's benefits in improved offloading decisions, resource utilization, and overall system performance outweigh the overhead-related latency. The proposed approach is designed to adapt to the dynamic nature of the vehicular environment and make real-time decisions based on the current system state, which helps mitigate the impact of overhead on the overall performance.

In conclusion, the overhead analysis quantifies the various overhead sources in the AIoT-DQN-PAVET approach. It separates the latency caused by the overhead from the protocol-related latency. Despite the additional overhead, the proposed approach still achieves lower latency compared to the benchmark schemes, demonstrating its effectiveness in handling the dynamic nature of VEC systems.

VI. CONCLUSION

The AIoT-enhanced DQN-based learning algorithm for priority-aware task offloading proposed in this article represented a significant advancement in VEC systems for vehicle road cooperation. By leveraging the power of augmented intelligence and deep reinforcement learning, the algorithm efficiently managed resources and prioritizes latency-critical tasks in dynamic vehicular environments. The theoretical contributions of this research, including the CMDP formulation and the development of the DQN-based algorithm, provide a solid foundation for future work in intelligent transportation systems and edge computing. The proposed algorithm has the potential to improve the performance and user experience of VEC systems significantly.

The main contributions of this article, including the novel AIoT-enhanced framework, the CMDP problem formulation, the DQN-based learning algorithm, and the extensive simulation experiments, demonstrate the effectiveness and potential of the proposed approach. The practical advantages of the algorithm, such as reduced latency, increased throughput, and improved resource utilization, directly benefit vehicle road cooperation applications and enhance user satisfaction.

However, the limitations of this study should be acknowledged, such as the reliance on simulation experiments, the need for further investigation into scalability, and the assumptions made regarding the availability and accuracy of real-time system state information. Future work should address these limitations through real-world experiments, exploration of scalability issues, and the development of robust methods for handling imperfect or incomplete system state information. Moreover, several limitations and challenges are encountered while implementing and simulating the proposed AIoT-enhanced DQN-based learning algorithm. One challenge is the computational complexity of training the DQN model, which increased with the size of the state and action spaces. This is addressed using techniques, such as experience replay and target network stabilization. Another area for improvement is the assumption of perfect knowledge of the system state, which may only sometimes be feasible in real-world scenarios.

Future research directions based on the findings and limitations of this study include extending the proposed framework to handle more complex scenarios, incorporating additional performance metrics into the optimization objective, and exploring the integration of other AI techniques to enhance the learning efficiency and adaptability of the proposed algorithm.

REFERENCES

[1] J. Dong et al., "Mixed cloud control testbed: Validating vehicle-road-cloud integration via mixed digital twin," *IEEE Trans. Intell. Veh.*, vol. 8, no. 4, pp. 2723–2736, Apr. 2023.
[2] B. Ko, K. Liu, S. H. Son, and K. J. Park, "RSU-assisted adaptive scheduling for vehicle-to-vehicle data sharing in bidirectional road scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 977–989, Feb. 2021.
[3] G. Yuan et al., "An autonomous vehicle group cooperation model in an urban scene," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 13852–13862, Dec. 2023.
[4] F. Jiang et al., "A new form of deep learning in smart logistics with IoT environment," *J. Supercomput.*, vol. 78, no. 9, pp. 11873–11894, Feb. 2022.
[5] C. Lin, G. Han, J. Du, T. Xu, L. Shu, and Z. H. Lv, "Spatiotemporal congestion-aware path planning toward intelligent transportation systems in software-defined smart city IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8012–8024, Sep. 2020.
[6] H. Jiang, X. Li, R. P. Mahapatra, and P. Poovendran, "Creating a ubiquitous learning environment using IoT in transportation," *Soft Comput.*, vol. 27, no. 2, pp. 1213–1213, Aug. 2021.
[7] S. T. Ahmed, V. V. Kumar, and J. Kim, "AITel: eHealth augmented-intelligence-based telemedicine resource recommendation framework for IoT devices in smart cities," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18461–18468, Nov. 2023.
[8] L. Zong, D. W. Qiao, H. Wang, and Y. Bai, "Sustainable cross-regional transmission control for the industrial augmented intelligence of Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10214–10223, Oct. 2023.

[9] G. Rathee, A. Kumar, S. Grag, B. J. Choi, and M. M. Hassasn, "ART: Active recognition trust mechanism for Augmented Intelligence of Things (AIoT) in smart enterprise systems," *Alex. Eng. J.*, vol. 80, pp. 417–424, Oct. 2023.

[10] L. Chen, L. Xue, H. Huang, W. Wang, M. Cao, and F. Xiao, "Double rainbows: A promising distributed data sharing in augmented intelligence of things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 653–661, Jan. 2023.

[11] D. Javeed, T. Gao, M. S. Saeed, and M. T. Khan, "FOG-empowered augmented-intelligence-based proactive defensive mechanism for IoT-enabled smart industries," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18599–18608, Nov. 2023.

[12] R. Mishra, H. P. Gupta, R. Kumar, and T. Dutta, "Leveraging augmented intelligence of things to enhance lifetime of UAV-enabled aerial networks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 586–593, Jan. 2023.

[13] B. Ducourthial, V. Cherfaoui, T. Fuhrmann, and S. Bonnet, "Experimentation of a road hazard anticipation system based on vehicle cooperation," *Veh. Commun.*, vol. 36, Aug. 2022, Art. no. 100486.

[14] J. Imbsweiler, M. Ruesch, H. Weinreuter, F. P. Leon, and B. Deml, "Cooperation behaviour of road users in t-intersections during deadlock situations," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 58, pp. 665–677, Oct. 2018.

[15] L. Ding et al., "Learning to help emergency vehicles arrive faster: A cooperative vehicle-road scheduling approach," *IEEE Trans. Mob. Comput.*, vol. 22, no. 10, pp. 5949–5962, Oct. 2023.

[16] M. Hasan, S. Mohan, T. Shimizu, and H. S. Lu, "Securing vehicle-to-everything (V2X) communication platforms," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 693–713, Dec. 2020.

[17] A. Alnasser, H. J. Sun, and J. Jiang, "Recommendation-based trust model for vehicle-to-everything (V2X)," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 440–450, Jan. 2020.

[18] B. M. ElHalawany, A. A. A. El-Banna, and K. Wu, "Physical-layer security and privacy for vehicle-to-everything," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 84–90, Oct. 2019.

[19] D. D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[20] Y. Sun et al., "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019.

[21] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and K.-L. A. Yau, "Virtual edge: Exploring computation offloading in collaborative vehicular edge computing," *IEEE Access*, vol. 9, pp. 37739–37751, 2021.

[22] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, Feb. 2020.

[23] J. Liu, N. Liu, L. Liu, S. Li, H. Zhu, and P. Zhang, "A proactive stable scheme for vehicular collaborative edge computing," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10724–10736, Aug. 2023.

[24] D. Chen, Y.-C. Liu, B. Kim, J. Xie, C. S. Hong, and Z. Han, "Edge computing resources reservation in vehicular networks: A meta-learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 10724–10736, May 2020.

[25] C. Sonmez, C. Tunca, A. Ozgovde, and C. Ersoy, "Machine learning-based workload orchestrator for vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2239–2251, Apr. 2021.

[26] R. Meneguette, R. De Grande, J. Ueyama, G. P. Rocha Filho, and E. Madeira, "Vehicular edge computing: Architecture, resource management, security, and challenges," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–46, Jan. 2023.

[27] S. Liu, J. Tian, C. Zhai, and T. Li, "Joint computation offloading and resource allocation in vehicular edge computing networks," *Digit. Commun. Netw.*, vol. 9, no. 6, pp. 1399–1410, Dec. 2023.

[28] Q. Shen, B.-J. Hu, and E. Xia, "Dependency-aware task offloading and service caching in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 13182–13197, Dec. 2022.

[29] H. Ko, J. Kim, D. Ryoo, I. Cha, and S. Pack, "A belief-based task offloading algorithm in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5467–5476, May 2023.

[30] Y. Xia, H. Zhang, X. Zhou, and D. Yuan, "Location-aware and delay-minimizing task offloading in vehicular edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16266–16279, Dec. 2023.

[31] L. Chen, Y. Ji, T. Xie, J. Ding, and J. Wan, "Cross-layer cooperative offloading in vehicular edge computing networks," *Veh. Commun.*, vol. 42, Aug. 2023, Art. no. 100624.

[32] D. Ngoduy, N. H. Hoang, H. L. Vu, and D. Watling, "Multiclass dynamic system optimum solution for mixed traffic of human-driven and automated vehicles considering physical queues," *Transp. Res. B, Methodol.*, vol. 145, pp. 56–79, Jan. 2021.

[33] A. Khezrian, T. D. Todd, G. Karakostas, and M. Azimifar, "Energy-efficient scheduling in green vehicular infrastructure with multiple roadside units," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1942–1957, May 2015.

[34] H. Ko, S. Pack, and V. C. M. Leung, "Coverage-guaranteed and energy-efficient participant selection strategy in mobile crowdsensing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3202–3211, Apr. 2019.

[35] X. Zhao, G. Q. Huang, J. Jiang, L. Gao, and M. Li, "Task offloading of cooperative intrusion detection system based on deep Q network in mobile edge computing," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117860.