Engineering Science and Technology, an International Journal 59 (2024) 101854

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

Deep reinforcement learning approach for multi-hop task offloading in vehicular edge computing

Manzoor Ahmed [a,b], Salman Raza [c], Haseeb Ahmad [c,*], Wali Ullah Khan [d], Fang Xu [a,b], Khaled Rabie [e,f]

[a] School of Computer and Information Science, China
[b] Institute for AI Industrial Technology Research, Hubei Engineering University, Xiaogan City 432000, China
[c] Department of Computer Science, National Textile University, Faisalabad, Pakistan
[d] Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, 1855 Luxembourg City, Luxembourg
[e] Department of Computer Engineering & Center for Communication Systems and Sensing at King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia
[f] Department of Electrical and Electronic Engineering Science at the University of Johannesburg, South Africa

ARTICLE INFO

Keywords:
Vehicular edge computing
Task offloading
Multi-hop
MmWave

ABSTRACT

The rise of Vehicular Edge Computing (VEC) has gained attention for its ability to alleviate backhaul network load and provide ultra-low latency. In meeting the escalating computational needs of cutting-edge vehicular applications such as augmented reality and autonomous driving, the abundant computational resources of vehicles can prove critical for task computation in a VEC environment. Nevertheless, the high mobility of vehicles has the potential to disrupt ongoing task computation due to varying communication network connectivity. This paper proposes a task offloading scheme that leverages multi-hop vehicle computation resources in Vehicle-to-Vehicle (V2V) communication, relying on mobility analysis. Vehicles capable of fulfilling the requisite communication and computation demands via multi-hop connectivity can assist in performing tasks offloaded by the client vehicle, along with the single-hop vehicles in the vicinity of the client vehicle. We formulate an NP-hard optimization problem for task offloading to minimize all tasks' weighted sum of computation delay. For this, a proximal policy optimization-based Multi-hop Vehicular Task Offloading (MVTO-PPO) scheme in vehicular edge computing is designed for low complexity that provides the optimal solution. Our approach involved modeling the task offloading process as a Markov decision process. We then developed an offloading decision algorithm that utilizes deep reinforcement learning to choose the appropriate vehicle for task execution. This approach improves the quality of environmental perception by enabling reasonable task offloading, ultimately leading to significant long-term benefits. Furthermore, we explore the integration of fifth-generation new-radio vehicle-to-everything (5G NR V2X) communication, utilizing both cellular links and millimeter wave technology to enhance system performance. Simulation results demonstrate that the proposed algorithm significantly reduces task offloading delays, outperforming benchmark approaches in various scenarios.

1. Introduction

The Internet of Vehicles (IoV) has gained substantial devotion in the recent past from the research community, [1–3], and has undergone significant development in conjunction with the emerging devices and technology [4]. This has resulted in significant advancements in various automotive applications to strengthen road safety, enhance traffic efficiency, and offer customers convenient entertainment services [5]. However, applications with stringent latency constraints and high processing needs, like autonomous driving, create significant demands for computing and storage resources [6]. Vehicular Edge Computing (VEC) is a potential strategy for meeting this requirement [7]. VEC helps reduce response time and relieve back-end network stress by relocating computing and storage resources near vehicular users [8]. The combination of Mobile Edge Computing (MEC) and vehicular networks

* Corresponding author.
E-mail addresses: manzoor.achakzai@gmail.com (M. Ahmed), salmanraza1214@gmail.com (S. Raza), haseeb_ad@hotmail.com (H. Ahmad), waliullah.khan@uni.lu (W.U. Khan), xf@hbeu.edu.cn (F. Xu), k.rabie@kfupm.edu.sa (K. Rabie).

https://doi.org/10.1016/j.jestch.2024.101854
Received 21 February 2023; Received in revised form 26 September 2024; Accepted 2 October 2024
Available online 21 October 2024

enables this possibility. However, due to the distinct characteristics of vehicular networks, such as high node/vehicle mobility and varying channel conditions, developing an effective approach for offloading tasks enabled by edge computing is a significant challenge.

A lot of attention has been given to offloading tasks in VEC, particularly in studying server-based offloading strategies [9]. These strategies involve using roadside units (RSUs) as edge servers with limited but adequate communication, computation, and storage capabilities. In [10], the scheduling of workloads was achieved on a single edge server, whereas [11,12] investigated task offloading based on mobility across multiple edge servers. Nevertheless, edge server penetration is minimal during the initial stages of implementation. Additionally, the implementation of edge servers might be fairly costly. Fully deploying edge servers along road sections is unfeasible from a practical standpoint, particularly on a large-scale network. In contrast, edge servers have trouble keeping up with the rapidly growing volume of data due to their lower computing capacity than the cloud.

The utilization of Vehicle-to-Vehicle (V2V) offloading presents an appealing approach to overcome the restrictions of server-based offloading and tackle the deficiency of computing capabilities in VEC. On the one hand, the number of automobiles continues to grow. On the other hand, vehicles are poised to become more intelligent, equipped with faster processors and increased storage capacity. As a result, these smart automobiles will be capable of communication and possessing computing and caching resources for task processing [13]. There is a significant opportunity to improve the capacity of automotive networks for communication and computation by exploiting the vast, underutilized resources in individual automobiles. We can boost efficiency and provide better customer service by pooling all these unused resources. Through resource sharing, V2V offloading enables proximity services that specifically benefit from proximity to vehicular users. To date, few efforts have been made to build an efficient V2V offloading system. While some works (for example, [14]) use vehicles' computational resources to provide services, they either fail to account for vehicle mobility or assume that all vehicles will voluntarily contribute their resources for free. Most current research on V2V offloading, in particular, only looks at vehicles that can finish tasks in a single-hop instead of the potential utilization of multi-hop vehicles.

Based on the earlier discussion, we must build an effective V2V offloading mechanism. In designing such a scheme, it is crucial to consider factors such as resource discovery, which can vary depending on each vehicle's mobility and computational capacity. However, the complex channel environment and dynamic topology make multi-hop links vulnerable. Because of this, it is very important to find vehicles that are easy to get for task execution. The constant changes in network topology due to the high mobility of vehicles pose a significant challenge to making optimal decisions. An efficient offloading process should also ensure that the tasks sent to its service vehicle are finished before the link between them is terminated, owing to the brief contact time between vehicles. Response time is the most demanding criterion to assess service quality since it requires computation and communication [15,16]. Thus, one of the primary objectives of an offloading method is to minimize response time. Likewise, because vehicle users are selfish, computing and storage resources may not be allowed unless compensated well. To promote resource sharing across vehicles, an incentive mechanism is required.

Our proposed Proximal Policy Optimization (PPO)-based Multi-hop Vehicular Task Offloading (MVTO-PPO) in VEC is to enhance resource utilization and improve user experience by maximizing unexploited vehicle resources. Its innovation lies in that the offloading policy is formulated by resolving the mobility issue generated by assigning tasks to single-hop service vehicles and utilizing the advantages of multi-hop vehicles in task computation. With single-hop vehicles, any given task must be finished before exiting the range of communication. In the case of multi-hop vehicles, participation in task computing is only possible when the link connectivity between the client and service vehicle is established prior to the former receiving the outputs of the computations. A mixed-integer program is developed in accordance with the proposed work offloading strategy. Considering the limitations imposed by connectivity on single-hop and multi-hop vehicles is critical for determining a workable solution. Last but not least, we added cutting-edge communication technology, which made the system more efficient by reducing the average time it took to unload.

Below is a brief overview of our paper contribution.

(1) A task offloading mechanism utilizing multi-hop vehicles that match the specified mobility pattern and computational capacity requirements is proposed. Our V2V offloading procedure considers many hops using a 5G-based millimeter wave (mmWave) communication mode, which enhances system performance by reducing offloading delays.
(2) To maximize long-term gains, the PPO is used to create task offloading decisions in the context of vehicular edge computing's Multi-hop Vehicular Task Offloading model.
(3) An offloading method is formulated by considering the offloading latency to minimize the client vehicle's utility while simultaneously complying with communication constraints. To ensure the effectiveness of our proposed method, we conduct exhaustive simulations. The experimental findings show that the suggested technique efficiently reduces task offloading latency compared to reference approaches.

Here is how the rest of the paper is structured. Section 2 covers vehicular task offloading-related works. We explain the system model in Section 3, and the problem formulation is explored in Section 4. Section 5 presents a multi-hop task offloading technique based on PPO. Section 6 contains the numerical results, and Section 7 concludes the paper.

2. Related work

In the VEC, vehicle users primarily use edge servers with specific computing and cache resources, such as RSUs, to perform task execution. The study in [17] examines workload offloading and job scheduling while accounting for vehicles' high mobility and time spent within the communication range of edge servers. To strike a middle ground between the needs of automobiles and edge servers, a dual-side optimization problem in [18] is proposed. In [1], a high-reliability, low-latency communication architecture for vehicles and infrastructure is presented. This design enhances wireless resource management and the coupling between vehicles and small base stations. According to [19], the task offloading strategy is meant to concurrently optimize the destination server's decision and the transmission technology selection. In [20], the authors address the resource by incorporating caching, networking, and computation using a deep reinforcement learning (DRL) approach.

There is still a significant challenge in using today's vehicles' enhanced storage, processing, and sensing capabilities. The V2V communication mode was used to develop a task offloading approach that took advantage of idle vehicles' resources [21]. The authors used the Max-Min fairness method to balance task completion time and the number of participating vehicles. However, this strategy does not evaluate the effect of vehicle movement on offloading efficiency. The relay task offloading approach, which uses the vehicle's unused resources, was introduced in [12] to lower total network offloading costs. In this solution, once again, V2V mode is used to offload tasks for execution, lowering the burden on the edge cloud. The availability of resources is highly unpredictable because a vehicle can enter or leave the communication range of neighboring automobiles at any time [8]. Therefore, scheduling the task to proceed while the vehicle is there is the most effective method for ensuring it gets done. The authors in [14] presented an effective task offloading method that cuts down on the total offloading cost of the system. To address this problem and alleviate the strain on the edge server, the authors utilized the computing resources of vehicles. However, when offloading client vehicle tasks, the authors failed to take into account multi-hop task offloading, which is a key issue from the task offloading viewpoint.

In [22], the authors approach computation offloading as a problem of optimizing multiple objectives within constraints. They introduce the application of the Non-dominated Sorting Genetic Strategy to achieve both local and edge parallel processing, thereby reducing delay and energy consumption. Additionally, Zhang et al. investigated task offloading in small cellular network scenarios in [23]. They harnessed the artificial fish swarm algorithm to optimize system energy while minimizing delay. In their study detailed in [24], the authors also proposed a vehicle cooperative communication method based on fuzzy logic and signal game strategy. This method enhanced the accuracy of vehicle position and speed information using Kalman filtering, and relay nodes were strategically selected to aid in distributing information messages using fuzzy logic. Their research was further extended in [25], where they applied the greedy boundary stateless routing method to meet the communication requirements of self-organizing vehicle networks. In another study, [26], Zhang et al. employed a greedy selection method to establish data return links by choosing neighbor nodes with the highest stability and efficiency as relay nodes. Addressing the challenge of unreliable links due to the rapid mobility of vehicles in clusters, the authors introduced a passive multi-hop clustering algorithm in [27]. This algorithm was designed to ensure both coverage and stability within clusters. Finally, considering computing tasks' instability, heterogeneity, and interdependence, [28] provided a genetic algorithm for task offloading. When communicating between automobiles, the authors did not consider advanced communication technology. The authors also consider single-hop transmission, which has serious downsides compared to multi-hop V2V communication [29]. In [30], Considering the impact of the different hop, the authors introduced the hop count k and selected the neighboring vehicles in the k -hop wireless communication range as the candidate vehicles. To solve this problem, the authors proposed the greedy and discrete bat algorithms, respectively.

DRL has emerged as the preferred approach to achieve exceptional performance in different vehicular environments. In [31] presented a DRL-based offloading strategy (DRLOS) for MEC systems. The strategy leverages DRL to optimize offloading decisions dynamically in real-time, balancing task offloading between local devices and MEC servers. The study demonstrates that DRLOS can effectively reduce energy consumption and improve the overall latency of task execution. However, suffers from high computational complexity, limiting its feasibility for resource-constrained devices. Additionally, the traditional Q-learning algorithm used is prone to dimensional disaster. In [32] proposed a federated Q-learning technique to minimize the computational and communication budget and offload failure probabilities by utilizing resource-rich vehicles as service vehicles. Sheng et al. [33] proposed a classification scheme for client and fog vehicles. The authors deployed machine learning and coded computing techniques for vehicular network delay-sensitive data offloading issues. The authors in [34] proposed a two-stage task offloading approach where tasks are offloaded to RSUs and then offloaded to vehicles. However, they did not consider V2V offloading modes for vehicles. In [35], authors introduced an edge caching method using multi-agent DRL to address response delay issues in IoV caused by increased data traffic. In [36], an incentive-based Q-Learning approach was suggested to enhance user privacy in perception, along with a system model that combines mobile crowd sensing with MEC. The existing literature highlights the impracticality of fixing the task offloading mode. Vehicles in an IoV scenario possess computing resources. Vehicles can produce computing tasks, which implies that they can provide or require task computing services at various times.

Table 1
Frequently used notations.

Symbols	Description
$\mathcal{G}_b^i(\phi)$	Function of steering angle
L_n^{V2V}	Represents the path loss
N	Set of task $N = \{1, ..., N\}$
M	Set of service vehicles $M = \{1, ..., M\}$
s_i	Input data size for task i
ω_i	Computational resource needed for computing of task i
P_j	The connectivity of the routing path j
$d_{n,i}(t)$	The remaining distance before departing the coverage area of the vehicles at time t
$t_{n,stay}^{V2V}$	Time that vehicle stays under the coverage area of other vehicles
T_k	Time required to process tasks on service vehicle k from client vehicle n
$T_{k,i}^u$	Time required to upload task i to single-hop vehicle k
$r_{h_{cc}}$	Average transmission rate among for cellular communications
$r_{h_{mm}}$	Average transmission rate among for mmWave communication

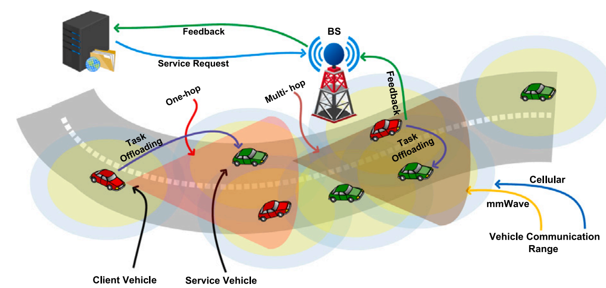


Fig. 1. Task offloading for VEC in the context of cellular and mmWave technologies.

3. System overview

We take into consideration a scenario depicted in Fig. 1 in which there are two directions of vehicles on a straight road (such as a highway). In this case, there are two vehicle types: those performing client tasks and those providing service. Service automobiles pool their unused computational resources to complete tasks generated by client vehicles. It is worth mentioning that the function of each vehicle can vary based on its mission or computational resources. Table 1 summarizes the most relevant symbols in this paper.

More than one function can be included in a single automotive app. For instance, as presented in [37], an augmented reality driving aid application consists of object identification and video streaming. The amount of data stored, the difficulty of computations, etc., are just a few of the characteristics of each service. Each service provided is viewed as a task. Both methods can be used to complete a client vehicle's tasks. When the client vehicle handles all of the processing for its tasks, computations are performed locally. In the second method, "offloading", the client vehicle sends its work to nearby vehicles with extra computing power.

To determine where and how to compute its tasks to meet its needs, the client vehicle must first gather information about the available service vehicles. Only then can it create an effective algorithm for task allocation. In this optimization problem, the best offloading decisions are determined by considering both the processing latency and the incentive pay. This includes wireless transmission delays for receiving and uploading data and vehicle computing delays. However, unlike with conventional MEC, with VEC, the total delay can be affected by the distance between the client and service vehicles. Motivating vehicles with surplus processing resources to take part in task offloading via some incentive system is crucial if we maximize vehicular resource usage and enhance vehicular users' experience. Client vehicles need to consider how much it will cost to use service vehicles for processing

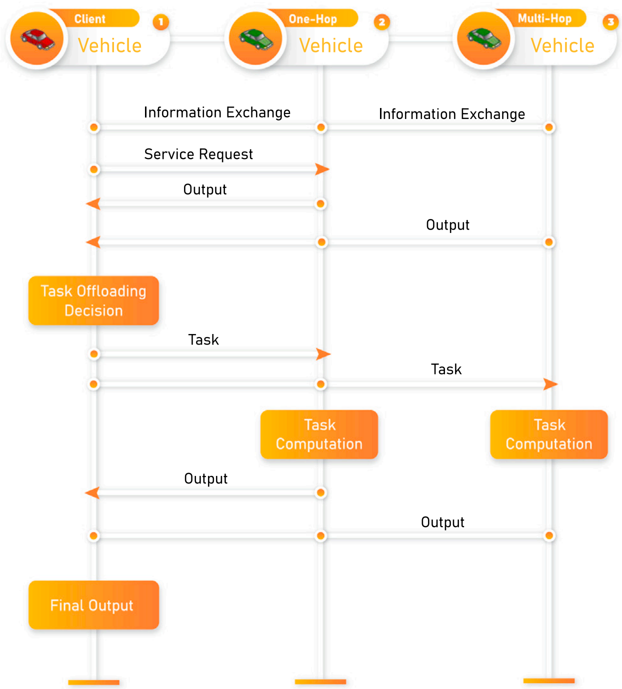


Fig. 2. The process of the proposed task offloading scheme.

tasks. It is the link connectivity that is crucial in ensuring offloading tasks. Links in-vehicle networks frequently fail because nodes are constantly moving. Consequently, the connectivity between the vehicles will significantly affect the distribution of the tasks. Better link connectivity makes it more likely that the client vehicle will give a task to its single-hop service vehicle.

The availability of links is a critical factor in the design of multi-hop service vehicles, as they are more prone to disruptions than single-hop vehicles. The offloaded tasks can be received by the multi-hop service vehicle from the client vehicle. The results can be sent back to the client vehicle if the connection between the two vehicles remains stable during the offloading procedure. The client vehicle can achieve the optimal offloading decision by addressing the posed challenge. When a client vehicle must rely on one or more service vehicles for task processing, it makes that determination through offloading decisions. Then, it sends the relevant tasks to the service vehicles in question. Each service vehicle will contribute to task processing once these tasks are received. Service vehicles complete their missions and report their results. The application can be run once the client has received all the related data. Referring to the flow chart in Fig. 2, it can be seen how the entire offloading strategy would work.

3.1. Mobility model

We assume that each vehicle moves at a random speed, where the speeds are independent and follow a Gaussian distribution. Since vehicle speeds vary over time, each vehicle is assigned a random speed υ, selected from a truncated Gaussian distribution to prevent negative velocities.

The truncated Gaussian probability density function (PDF) is expressed as:

$$f_{\upsilon}^{\sim} = \frac{2 f_{\upsilon}(\upsilon)}{erf(\frac{V_{max}-\mu}{\vartheta\sqrt{2}}) - erf(\frac{V_{min}-\mu}{\vartheta\sqrt{2}})}, \tag{1}$$

where $f_{\upsilon}(\upsilon) = \frac{1}{\vartheta\sqrt{2}} exp(-\frac{(\upsilon-\mu)^2}{2\vartheta^2})$ is the Gaussian PDF, $V_{max} = \mu + 3\vartheta$ is the maximum velocity, and $V_{min} = \mu - 3\vartheta$ is minimum vehicular velocity, $erf(.)$ is error function, μ is the average speed, and ϑ is defined as a standard deviation of vehicular speed. Therefore, based on (1), a vehicle speed within ($V_{min} \le \mu_{\upsilon} \le V_{max}$) is expressed as:

$$\mu_{\upsilon} = \frac{1}{\int_{V_{min}}^{V_{max}} \frac{f_{\upsilon}^{\sim}(\upsilon)}{\upsilon} d\upsilon} = \frac{erf(\frac{V_{max}-\mu}{\vartheta\sqrt{2}}) - erf(\frac{V_{min}-\mu}{\vartheta\sqrt{2}})}{\frac{2}{\vartheta\sqrt{2\pi}} \int_{V_{min}}^{V_{max}} \frac{exp(-\frac{(\upsilon-\mu)^2}{2\vartheta^2})}{\upsilon} d\upsilon}, \tag{2}$$

where μ and ϑ are derived as described in [1].

3.2. Communication model

V2V communication uses 5G NR-V2X Mode-2's cellular and mmWave connectivity. Each vehicle includes several antennas for mmWave and 5G connectivity. Vehicle distance determines communication mode. Cellular and mmWave modes are detailed below.

3.2.1. Cellular mode

The NR-V2X Mode-2 protocol handles the cellular link in V2V communication. In light of this, the transmission rate between vehicles υ_n and the vehicle υ_i can be calculated as follows:

$$R_{cc}^{\upsilon_n,\upsilon_i} = W_{cc} log_2(1 + \frac{P_t L_n^{V2V} |h^2|}{N_0}). \tag{3}$$

where P_t denotes the transmission power of the vehicle, the channel fading coefficient is represented by h, and the additive white Gaussian noise power is represented by N_0. While $W_c c$ and L_n^{V2V} represent the bandwidth and path loss, respectively, they are calculated according to [29] as follows:

$$L_n^{V2V} = 10^{-\frac{63.3+17.7log_{10}(d_{n,i})}{10}}, \tag{4}$$

where the $d_{n,i}$ implies the distance between vehicles υ_n and υ_i.

In addition, the length of time that vehicle υ_n spends within vehicle υ_i's coverage area is a crucial metric to analyze to prevent offloading failure. Using this, we can calculate the remaining distance before vehicle υi leaves its coverage area at time t.

$$d_{n,i}(t) = \sqrt{r_i^2 - (x_i(t) - x_n(t))^2} \pm (y_i(t) - y_n(t)), \tag{5}$$

where positions of vehicles υ_n and υ_i at time t are represented by $x_n(t), y_n(t)$ and $x_i(t), y_i(t)$, respectively, and r_i denotes the radius of vehicle υ_i's coverage area. The time that vehicle υ_n remains within the coverage area of vehicle υ_i can be calculated as:

$$t_{n,stay}^{V2V} = \frac{d_{n,i}(t)}{|\vec{\upsilon}_n - \vec{\upsilon}_i|}, \tag{6}$$

where $|\vec{\upsilon}_n - \vec{\upsilon}_i|$ represents the difference in the speeds of υ_n and υ_i with respect to their relative direction. This means that the rate of transmission, denoted by $R_{cc}^{\upsilon_n,\upsilon_i}$ varies over time, as indicated by $R_{cc}^{\upsilon_n,\upsilon_i}(t)$. Accordingly, the average rate of transmission between υ_n and υ_i is therefore given as:

$$r_{h_{cc}} = \frac{\int_0^{t_{n,stay}^{V2V}} R_{cc}^{\upsilon_n,\upsilon_i}(t)dt}{t_{n,stay}^{V2V}}. \tag{7}$$

3.2.2. mmWave mode

In order to take advantage of mmWave in V2V communication mode, we assumed that directional antennas would be mounted in each vehicle. Therefore, the vehicles' antennas must be aligned before the task transfer can begin using mmWave communications. As a result, we can express the gain of the antenna on the vehicle i as a function of the steering angle as phi, and the general mmWave antenna gains $\mathcal{G}^i b(\phi)$ are stated by [38] as:

$$\mathcal{G}_b^i(\phi) = \begin{cases} \mathcal{G}_i^{max}, & if\ |\phi| \le \phi_b \\ \mathcal{G}_i^{min}, & otherwise\,. \end{cases} \tag{8}$$

Mathematical gains for the main lobe and side lobe are $\mathcal{G}^i max$ and $\mathcal{G}^i min$, respectively, where phi denotes the angle off the boresight direction. The beam width of the major lobe is denoted by ϕb. Additionally, $R_{mm}^{v_n,v_i}$ derives the transmission rate of the vehicles v_n and v_i as:

$$R_{mm}^{v_n,v_i} = W_{mm} \log_2(1 + SNR_{v_n,v_i}), \tag{9}$$

where $R_{mm}^{v_n,v_i}$ is defined as:

$$r_{h_{mm}} = \frac{\int_0^{t_{n,stay}^{V2V}} R_{mm}^{v_n,v_i}(t)dt}{t_{n,stay}^{V2V}}, \tag{10}$$

where signal-to-noise ratio (SNR) of mmWave based V2V link is estimated according to [39] is obtained as:

$$\begin{aligned} SNR_{v_n,v_i} &= P_i - N_o - 10\log_{10}(W_{mm}) + 2\mathcal{G}^{main} \\ &\quad - 10\varphi \log_{10}(d_{n,i}) - 69.6 - \rho_\alpha \end{aligned} \tag{11}$$

Vehicle transmit power for mmWave is denoted by P_t, noise power in mmWave links is denoted by N_o, mmWave bandwidth is denoted by W_{mm}, path loss exponent is denoted by $varphi$, and distance between v_n and v_i is denoted by $d_{n,i}$ under the mmWave coverage of vehicles [14]. The main lobe array gain, $\mathcal{G}^{main}$, is tuned to 13 dB, and the shadow fading, ρ_α, a log-normal random variable with a zero-mean standard deviation, is given a value of 3 dB for LoS [38].

3.3. Task model

Considering an N-task client vehicle denoted by the set of N tasks as $N = \{1, \dots, N\}$. Additionally, let $M = \{1, \dots, M\}$ define the set of available service vehicles. Each task $i \in N$ is represented by a tuple $D = s_i, \omega_i$, where s_i corresponds to the input data size, and ω_i denotes the number of CPU cycles necessary for the completion of the task. The client vehicle, denoted as o executes its task locally or remotely. The offloading variable for job i and $k_\subseteq M$ is denoted as $x_{ki} \in \{0, 1\}$, where $x_{0i} = 1$ implies that task i is computed locally, while $x_{0i} = 0$; otherwise, $x0i = 0$. Similarly, $x_{ki} = 1$ indicates that task i is offloaded to it corresponding service vehicle k, while $x_{ki} = 0$; otherwise, $xki = 0$.

We provide a performance study of each processing approach, including response delay and computation cost, to aid in making effective offloading decisions. In addition, service constraints are included to account for the impact of mobile vehicles on task offloading.

3.3.1. Local computation

To illustrate, we will use f_n to represent the client vehicle o's local computing capacity regarding CPU cycles per second. For client vehicle n, the time needed to do task i in its immediate vicinity is then:

$$t_{n,i} = \frac{\omega_i}{f_n}. \tag{12}$$

According to (13), the time needed to process all tasks that must be executed locally can be derived as follows:

$$t_n = \sum_{i=1}^{N} x_{n,i} t_{n,i}. \tag{13}$$

3.3.2. Vehicle offloading

Specifically, we distinguish between single-hop and multi-hop service vehicles as they pertain to client vehicle O. For clarity, we will refer to the sets of vehicles that can make single hop $S_{one}(n)$ and those that can make multiple hops $S_{mul}(n)$ separately. Client vehicle O can learn about its single-hop and multi-hop vehicles' characteristics, such as location, velocity, and computational capability owing to V2V communication. Client vehicle n can use the data gathered to develop a useful offloading policy for single- and multi-hop vehicles.

In terms of client vehicles, the total amount of time it takes to complete task i for every given service vehicle k is the time interval $T_{k,i}$ consists of the three components, i.e., the transmission time $T_{k,i}^u$, execution time $T_{k,i}^c$, and output time $T_{k,i}^f$.

$$T_{k,i} = T_{k,i}^u + T_{k,i}^c + T_{k,i}^f. \tag{14}$$

The time required to process tasks on service vehicle k from client vehicle n can be expressed as:

$$T_k = \sum_{i=1}^{N} x_{k,i} T_{k,i}. \tag{15}$$

3.3.2.1. Single-hop transmission time. The time needed to upload task i from client vehicle to its single-hop vehicle $k \in S_{one}(n)$ is given by:

$$T_{k,i}^u = \frac{S_i}{r_h}, \tag{16}$$

where $r_h = \{r_{h_{mm}}, r_{h_{cc}}\}$ is the selected transmission rate according to Algorithm 1.

Due to vehicle mobility restrictions, tasks transferred from client vehicle n to single-hop service vehicles must be completed before the client vehicle and associated service vehicle's link becoming disconnected. Thus, computational offloading is successful if it is done before the link between two vehicles is lost. The time it takes for a client vehicle, n, to connect with its single-hop vehicle, k, is denoted by $t^{V2V} n, k, stay$. If the following equation holds, then a single-hop vehicle k can finish all the assigned tasks:

$$\sum_{i=1}^{N} x_{k,i} T_{k,i} \le t_{n,i,stay}^{V2V}. \tag{17}$$

3.3.2.2. Multi-hop transmission time. The intermediate relay nodes can deliver the data received by the client vehicle n to any multi-hop vehicle $k \in Smul(o)$. Let us assume H is the set of all possible multi-hop routing paths from client vehicle n to the vehicle k present in its multi-hop. Each multi-hop path $J \in H$ is presumed to contain u single-hop links connecting the client vehicle n to its multi-hop vehicle k. As a result, the connectivity of the j routing path can be represented as:

$$P_j = \min\{t_{n,1,stay}^{V2V}, t_{n,2,stay}^{V2V}, \dots, t_{(u-1),k,stay}^{V2V}\}. \tag{18}$$

Consider the path $n \to a \to b \to k$ where the numbers denote the link connections. If so, then this route has a connectivity of 3. We pick the H routing path with the highest connectivity for uploading tasks.

3.3.2.3. Task computation. The time required for service vehicle k to complete task i is given by:

$$T_{k,i}^c = \frac{\omega_i}{f_k}. \tag{19}$$

This analysis does not consider the time required for result feedback since the output size is significantly smaller than the input data.

4. Problem formulation

A multi-hop vehicular task offloading problem is defined as the client vehicle's utility minimization problem to improve the client vehicle's QoE. This problem is given by:

$$\begin{aligned} \textbf{P1:} \quad & \min_{x,T} \sum_{i=1}^{N} x_{k,i} T_{k,i} \\ s.t. \quad & C1: \sum_{k=0}^{M} x_{k,i} = 1, \quad \forall i \in N \\ & C2: x_{k,i} \in \{0,1\}, \quad \forall i \in N, k \in \{0, M\} \\ & C3: \sum_{i=0}^{N} x_{n,i} T_{n,k} \le t_{n,k,stay}^{V2V}, \quad \forall k \in S_{one}(o) \\ & C4: \sum_{i=0}^{N} x_{k,i} T_{k,i} \le P_{n,k}, \quad \forall k \in S_{mul}(o). \end{aligned} \tag{20}$$

The constraints on P1 are as described in the following: C1 indicates that each job can only be calculated at a single vehicle; C2 indicates that each job can be handled locally or via offloading; and C3 suggests that for successful task offloading, the processing time of the single-hop service vehicle must be less than or equal to the connection time between them where $P_{n,k}$ shows the connectedness of the client and multi-hop service vehicle routing path for data transmission that is determined using (18).

5. Multi-hop vehicular task offloading

In the context of a vehicular network, vehicles play a vital role in facilitating information exchange by periodically emitting beacons that convey details about their speed and current locations. Within this network, candidate service vehicles are defined as those positioned within a single-hop distance from a client vehicle's present location. This proximity is established through the client vehicle's consistent communication with nearby single-hop vehicles, enabling efficient task transmission between them. It is important to emphasize that vehicles located beyond a single-hop distance from the client's location can also be utilized for service. A multi-hop transmission mechanism is employed to enable communication between the client vehicle and these more distant service vehicles. This mechanism relies on intermediary vehicles acting as relay nodes, effectively bridging the gap between the client and remote service vehicles.

A significant constraint is that the inter-vehicle distance limits direct communication between the client and multi-hop vehicles. To overcome this limitation, the system leverages intermediary vehicles as relays. These relay nodes facilitate communication and provide critical information about the multi-hop vehicles, including their computational capabilities, positions, and speeds. As part of the task execution process, the client vehicle transmits its service requirements, encompassing factors such as desired delay thresholds and energy consumption limits. Single-hop vehicles play a pivotal role in this process, receiving the client vehicle's service requirements and forwarding them when the predefined connectivity parameters are met. When a single-hop vehicle meets the specified service standards, it can inform the client vehicle of its readiness to serve as the service vehicle for the return journey. This iterative process continues until the maximum allowable number of hops is reached. It is worth noting that multiple routing paths may exist between a service vehicle and a client vehicle, underscoring the importance of identifying the route that offers the highest throughput performance to establish the optimal data transmission path. With the data obtained through these interactions, the client vehicle gains insights into the availability of multi-hop vehicles. If a multi-hop vehicle is designated as the service vehicle, it initially receives tasks from the client vehicle through intermediary relay nodes before directly processing the assigned tasks.

5.1. Markov decision process model

The task offloading procedure used in the automotive context exhibits the Markov property. In other words, only the system's current state can determine its next state. The task offloading can be conceptualized as a Markov decision process (MDP). The PPO algorithm is a recent development in reinforcement learning algorithms. It benefits from a streamlined training procedure and efficient data utilization. So, we have decided to use the PPO algorithm to make offloading decisions in a V2V scenario.

MDP refers to a process in which an agent interacts with its surroundings. In this setup, the client vehicle acts as the agent and decides which tasks to give to other vehicles. The environment, which includes both client and service vehicles, is the road network as a whole.

There are distinct time intervals for scheduling purposes in the system. In order to make an offloading choice, the agent first needs to receive the state for this scheduling time from its environment. When the offloading decision is put into action, the surrounding environment will undergo a shift, and the agent will receive feedback on the consequences of this shift as a reward. Once the external conditions have changed, a new scheduling period begins, and the agent repeats the aforementioned steps until the system terminates. Here, we will look in more detail at the system's state, action, and reward function.

Algorithm 1: The Offloading Decisions with Communication Perspective

Input: $D = \{s_i, \omega_i\}, d_{n,i}(t), t^{V2V}_{n,stay}, r_{h_{cc}}, r_{h_{mm}}$
Output: r_h

while $t \le end$ **do**
  $r^t_h = r_{h_{cc}}$
  while $T_{k,i} \le t^{V2V}_{n,stay}$ **do**
    if *Distance of* v_i *and* $v_n \le$ *mmWave range (calculated using the equations* (1) *and* (2)*)* **then**
      Output transmission by mmWave mode
      $r^t_h = r_{h_{mm}}$
    else
      if *Distance of* v_i *and* $v_j \le$ *mmWave range* **then**
        Output transmission by mmWave mode via v_i to v_j then v_n in multi-hop
        $r^t_h = r_{h_{mm}}$
      else
        Output transmission by cellular mode via v_i to v_j then v_n in multi-hop
        $r^t_h = r_{h_{cc}}$
      end
    end
  end
  $t = t + 1$
end
$r_h = r^t_h$

5.1.1. State space

The learning agent bases its decision to delegate a task on the system state, which should account for all variables that could influence the profitability of the decision. For the system to function, the agent must base their decisions on the information of available vehicle communication resources and the tasks that need to be finished during the present scheduling period. When delegating tasks to a service vehicle, keeping the maximum range of communication is vital. The distance between vehicles, as stated in (5), and the communication method, mmWave or cellular, as defined in (7) and (10), respectively, decide whether the vehicles can communicate.

The system's state in period τ is represented as $S(\tau) = \{D'_\tau, M'_\tau, r'_h, W_\tau, p\tau\}$, where D'_τ is the set of all offloading demands for this period, M'_τ is the set of all available vehicle servers in this period. Total communication bandwidth in period τ is designated by W_τ, each vehicle's transmit power is denoted by p_τ, and the communication links availability between clients and service vehicles is expressed by the r'_h indicator matrix.

5.1.2. Action space

The agent's ability to decide whether or not each duty should be offloaded and to which vehicle constitutes the system's action space, denoted by $x(\tau)$. Task X_n offloading decision is given as:

$$x_k = (x_1, x_2, \dots, x_n), \tag{21}$$

where x is limited to either 0 or 1. The set of decisions regarding offloading for all tasks in period τ can be denoted by x_τ:

$$x_\tau = (x_1, x_2, \dots, x_K). \tag{22}$$

Algorithm 2: The MVTO-PPO Algorithm

Input: N, T, M, r_h
Output: θ, ω
The parameters of the strategy π_θ and the critic network, θ and ω, are initialized randomly.
;
Set the parameters of the strategy to θ', where $\pi_{\theta'} = \theta$;
while $i < N$ **do**
  Employ the strategy $\pi_{\theta'}$ to interact with the environment over a duration of T time steps, where vehicular speeds follow according to equations (1) and (2), and accumulate interaction data comprising $\{x_\tau, s_\tau, r_\tau\}$.
  $x(\tau) = \sum_{\tau'=\tau}^{\tau+n} \gamma^{\tau'-\tau} * r_{\tau'} - V^{\pi\theta}(s_\tau, \omega)$ is computed in each step
  Update the parameters of $\pi_{\theta'}, \theta' = \theta$
  while $j < M$ **do**
    Use the gradient descent method for optimizing θ and ω in accordance with (24)
  end
end

If the service vehicle that is supposed to compute a task is out of range with the client vehicle that originally had the task, then the task is not considered to have been offloaded.

5.1.3. Reward function

The agent's learning must be directed by the reward function. The reward needs to be in line with the objective of system optimization. This system's optimization objective is to reduce task offloading latency and enhance the client vehicle's QoE in a multi-hop vehicular environment.

The aggregate value of all servers' $T_{k,i}$ for the given time interval τ. At the time τ, the reward is determined by the ratio of T_τ to T_τ^{sum}, which considers the total amount obtained by the service vehicles to the combined sum of all tasks that the service vehicles attempted to offload at that time. Additionally, the discounted cumulative reward attained by the agent from the system's initial state to its final state is determined as follows:

$$Reward = \sum_{\tau=0}^{\tau_{end}} \gamma^\tau r(\tau), \tag{23}$$

where γ is the reward's discount coefficient and τ_{end} represents the time step of system's termination state.

5.2. Offloading decision algorithm based on PPO

Policy-based and value-function-based RL is used in the proximal policy optimization (PPO) algorithm.

In autonomous driving's task-offloading scenario, the strategy network (or actor-network) must engage with the surroundings to learn the task-offloading method that maximizes overall advantages. Agent's probability of making offloading decision x_τ in state s_τ is represented by $\pi(x_\tau|s_\tau)$, where θ are network parameters. The critic network assigns scores to the actions produced by the actor-network to assess their quality. The critic network is responsible for evaluating whether an action is beneficial or detrimental, and to do so accurately, it must estimate the value of $V^{\pi_\theta}(s, \omega)$ for each state under the strategy π_θ. This value represents the expected cumulative reward achieved by applying the strategy π_θ and examining the state x, where ω denotes the critic network's parameters. For the actor-network to produce the optimal action for a particular state, the critic network must know the optimal values for both θ and ω''.

The PPO algorithm is implemented by employing a critic network and two strategy networks, which correspond to the strategies π_θ and $\pi_{\theta'}$. When gathering information about how a given system interacts with its surroundings, $\pi_{\theta'}$ is the strategy of choice. The interaction data for a given time τ comprises the agent's action x_τ and the state $s\tau$ it occurred in, as well as the reward r_{tau} it received. The targeting strategy, $\pi\theta$, should be maximized. The objective of the critic network's optimization is to reduce the error in its evaluation of the state value, which involves minimizing the following function:

$$L^v(\omega) = E_\tau[V_{s_\tau} - V^{\pi\theta}(s_\tau, \omega)]^2. \tag{24}$$

The expression V_{s_τ} refers to the actual value of state s_τ, whereas $V^{\pi\theta}(s_\tau, \omega)$ is the estimated value of state s_τ provided by the critic network. In this context, $E_\tau[\cdot]$ denotes the expected value. Since directly obtaining the true value V_{s_τ} is not feasible. The PPO algorithm relies on the accumulated discounted reward gathered using n-step sampling after the state s_τ to estimate the true value of state s_τ, which is represented as:

$$V_{s_\tau} = \sum_{\tau'=\tau}^{\tau+n} \gamma^{\tau'-\tau} * r_{\tau'}, \tag{25}$$

where γ represents the discount factor used for the reward. The proposed algorithm's overall operation is explained in Algorithm 2.

5.3. Computational complexity

The computational complexity of Algorithm 1 is influenced by two main factors: the decision-making process based on distance and the communication mode used (cellular or mmWave). The most complex part of this algorithm involves checking the distance between vehicles and switching between cellular and mmWave transmission modes. For each iteration, the distance is checked, and based on that, either direct transmission (one-hop) or multi-hop transmission is decided. The complexity of calculating distances is $O(n)$ for checking each vehicle's distance from others, where n is the number of vehicles. The selection of the transmission mode adds a constant overhead, resulting in an overall computational complexity of $O(n)$.

The computational complexity of Algorithm 2 is more complex due to its DRL component. The complexity primarily depends on the number of neural network parameters and the number of iterations required for convergence. In each iteration, the policy gradient and value function updates are calculated, which involves forward and backward passes through the network. If the network has L layers and each layer contains k_l neurons, the complexity of one forward pass is $O(\sum_{l=1}^{L-1} k_l k_{l-1})$. Since we assume the number of neurons in each layer is the same, the complexity simplifies to $O(k^2)$, where k is the number of neurons in each layer. Therefore, the overall complexity of the MVTO-PPO algorithm per iteration is $O(k^2)$.

When combining Algorithms 1 and 2, the overall complexity is dominated by the more computationally intensive Algorithm 2. The complexity of Algorithm 1 is $O(n)$, which is significantly lower than the $O(k^2)$ complexity of Algorithm 2. Therefore, the combined complexity of the two algorithms is $O(k^2)$, as the neural network computations in Algorithm 2 overshadow the distance calculations in Algorithm 1.

6. Numerical results

This section contains numerical results as well as a discussion of the proposed algorithm. The simulation scenario under consideration is a one-way straight road. Along this road, vehicles are evenly distributed. Assume the client vehicle as a reference point whose location is an original point at a 60 km/h speed. Single-hop and multi-hop vehicles are considered service vehicles, with speeds ranging from 40 km/h to 60 km/h.

We assume beacon messages are broadcasted by vehicles to other vehicles that are present in their communication range, including computing resource information [40]. We used a mmWave connection in our case since message sizes are small compared to the greater

Table 2
Simulation parameters

Parameter	Value
V2V mmWave bandwidth W_{mm}	200 MHz
V2V cellular bandwidth W_{cc}	20 MHz
Client vehicle's CPU speed	400 MHz
Service vehicle's CPU speed	800–1600 MHz

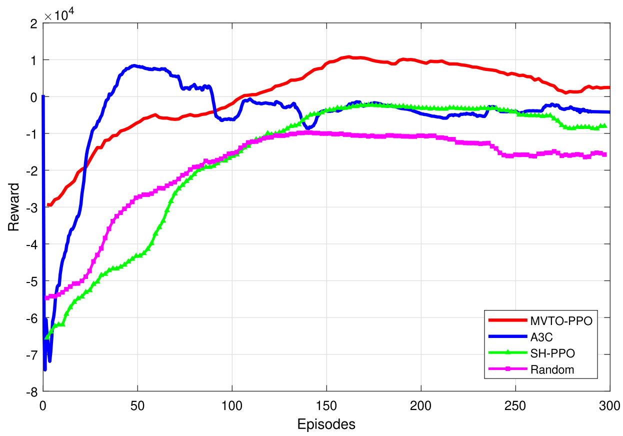


Fig. 3. Learning curves.

bandwidth used in 5G NR-V2X. Therefore, we ignore communication overhead. The other simulation parameters are summarized in detailed settings in Table 2.

The proposed task offloading scheme is evaluated against the following baseline schemes:

(1) Asynchronous Advantage Actor-Critic (A3C): The A3C algorithm is implemented to make the task offloading decisions. Service vehicles prioritize tasks based on their deadlines and follow the first-come-first-served method when discarding tasks unlikely to be completed on their deadline [41].
(2) Single-hop Proximal Policy Optimization (SH-PPO): The primary difference between SH-PPO and the proposed scheme is that SH-PPO only considers single-hop service vehicles.
(3) Random: A random set of task offloading decisions is adopted, meaning the tasks are randomly assigned to the service vehicles.

Fig. 3 illustrates the learning curves of the training process, which deduced that our proposed MVTO-PPO outperforms the A3C algorithm, SH-PPO, and Random schemes regarding convergence outcomes. The convergence speed shows that the proposed algorithm-based technique converges faster and more smoothly compared to the A3C algorithm and other methods. The proposed algorithm achieves greater task utilization and makes the learning process more stable than the algorithms by pruning the objective function.

Fig. 4 illustrates the impact of task computation intensity on rewards for various offloading algorithms. As computation intensity increases, all algorithms show a decline in rewards, primarily due to the increased task computation delay, which leads to longer total processing times and subsequently lower rewards. Among the evaluated methods, the MVTO-PPO algorithm consistently demonstrates superior performance, maintaining the highest rewards across all intensity levels. This is followed by A3C and SH-PPO, while the Random algorithm consistently exhibits the lowest performance. The MVTO-PPO's advantage lies in PPO's stability, robustness, and efficiency, enabling smoother updates and faster convergence compared to A3C. Additionally, the multi-hop capabilities of MVTO-PPO further enhance its effectiveness in environments with varying computational demands,

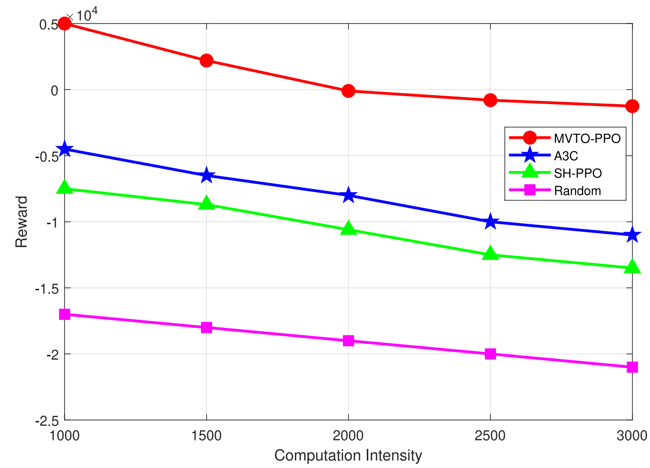


Fig. 4. Reward vs. Computation intensity for different algorithms.

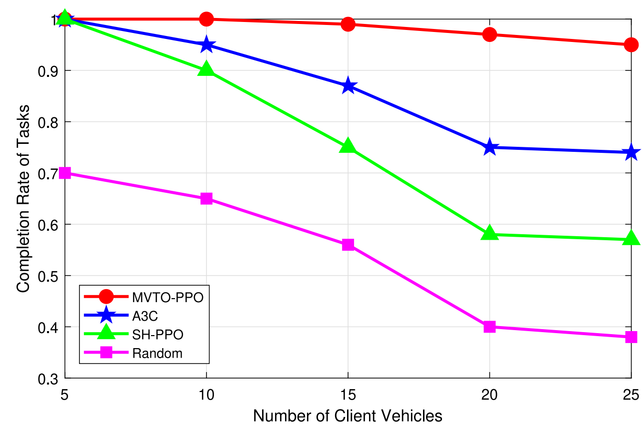


Fig. 5. Relationship between the tasks' completion rate and client vehicles.

making it more adaptable and efficient under increased workload conditions. This comparison underscores the robustness of MVTO-PPO in managing complex, computation-heavy tasks more efficiently than the other evaluated algorithms.

Fig. 5 depicts the task completion rate with an increase in the number of client vehicles. It can be observed from Fig. 5 that as the number of tasks increases, the tasks' completion rate decreases considerably in the A3C, SH-PPO, and Random schemes, while it always stays at almost 100% under the proposed MVTO-PPO scheme. Unlike the MVTO-PPO scheme, since the tasks are not suitably scheduled at the service vehicles, all schemes suffer from a considerable decrease in task completion rate as the number of tasks increases.

Fig. 6 shows the offloading delay when there are varying numbers of tasks in all schemes. An increase in tasks leads to a longer delay. A minimization of delay can be obtained with the cooperation of service vehicles. Nevertheless, a vehicle is selected randomly to perform a task in a random scheme. The optimal offloading decision failed to be generated by a random strategy, resulting in a longer delay than other schemes. In contrast, MVTO-PPO enhances resource utilization by considering both single- and multi-hop vehicles to complete tasks offloaded by the client vehicle. Thus, the obtained offloading decisions enable the fast computation of tasks.

The CPU cycles/bits are displayed by using the average delay of all methods with different computation intensities in Fig. 7. It can be observed that the trend depicted in Fig. 6, where an increase in the number of tasks results in an increase in delay, is similar to the trend of the delay experienced by the client vehicle in computing the tasks.

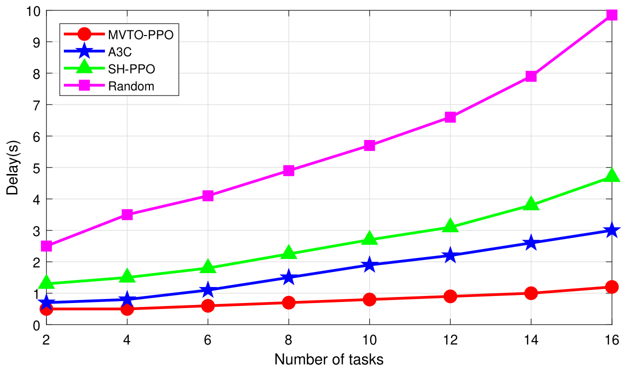


Fig. 6. Delay analysis while varying the number of tasks.

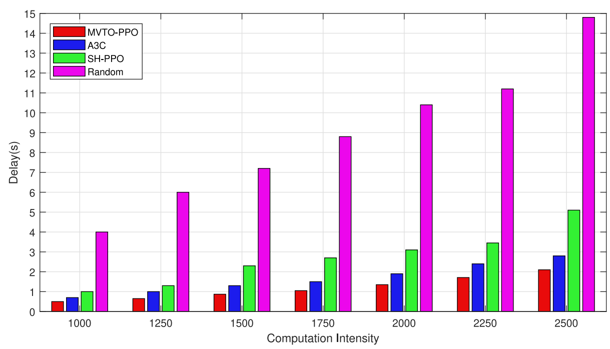


Fig. 7. Delay analysis while varying the computation intensity.

This delay also increases with the rise in computation intensity. As computation intensity increases, this phenomenon happens; the task computation needs more computing resources, resulting in a delay. When the computation intensity is increased from 1000 to 2500 cycles/bits, it can be observed that the MVTO-PPO scheme outperforms other schemes in terms of delay. Our proposed technique enables the client vehicle to leverage the benefits of offloading tasks to nearby single-hop and multi-hop service vehicles and make optimal offloading decisions using PPO, resulting in better performance. Therefore, the slightest delay variation occurs in case of increasing compute workloads.

In Fig. 8(a), we employ the MVTO-PPO algorithm. We choose service vehicles from neighboring vehicles at one hop, two hops, three hops, and four hops for task offloading. These selections visually represent the relationship between the number of tasks and the corresponding completion time. The figure in 8(a) underscores a significant pattern: an increase in hops corresponds to a reduction in execution delay. This finding robustly endorses the idea that multi-hop task offloading grows progressively advantageous as the number of hops increases. Fig. 8(b) provides a comparative analysis of service vehicles operating within a four-hop range. Specifically, we examine the performance of the MVTO-PPO, A3C, Random algorithms, and the one-hop SH-PPO algorithm. As the number of tasks increases, there is a corresponding rise in the total delay required to complete these tasks, which is influenced by transmission and computation delays. The client vehicles offload tasks to service vehicles using a multi-hop approach. The increase in the number of tasks naturally results in longer delays. The minimization of delay obtained on effective cooperation of service vehicles. However, the random selection of vehicles for task offloading, as seen in the Random scheme, falls short, resulting in longer delays than other schemes. Moreover, the SH-PPO scheme underperforms due to its limitation to single-hop task offloading. Even the A3C scheme encounters challenges, notably high variance, which can slow down convergence. In contrast, MVTO-PPO stands out by enhancing resource utilization by considering four-hop service vehicles for task completion. This strategic choice in task offloading decisions accelerates task computation, contributing to a notable enhancement in overall performance.

Fig. 9 illustrates the delay analysis with varying vehicular velocity. Regarding how speed affects performance, we can see from Fig. 9 that the offloading delay rises as the vehicular velocity increases. Due to the strict transmission delay constraint, the number of qualified vehicles drops down in single-hop and multi-hop together with the vehicle velocity. As a result, a vehicle is less likely to fulfill task requirements. The MVTO-PPO scheme is, therefore, more suited for all varying vehicle velocities.

6.1. Discussion

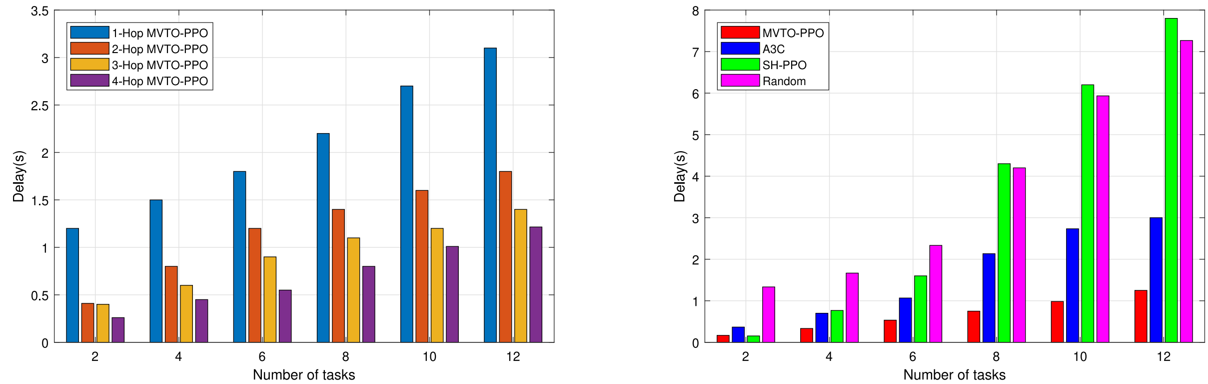
The evaluation of DRL performance, as presented in Figs. 3–8, highlights the superiority of the MVTO-PPO scheme over other DRL algorithms in VEC scenarios. The exceptional performance of MVTO-PPO is attributed to several key factors, including its integration of the state-of-the-art PPO algorithm, which is known for its stability, robustness, and computational efficiency. Unlike algorithms such as A3C, which may suffer from high gradient variance due to asynchronous updates from multiple agents, PPO ensures a smoother learning curve and faster convergence, striking a balance between sample efficiency and computational resource usage. This makes PPO particularly well-suited for complex, dynamic environments like multi-hop vehicular networks, where task offloading demands strategic decision-making.

Moreover, the MVTO-PPO scheme's multi-hop vehicular task offloading model introduces significant advancements in handling inter-vehicle communication and mobility patterns. By incorporating a multi-hop offloading mechanism, the scheme optimally manages task distribution across multiple vehicles, minimizing latency and maximizing resource utilization in challenging vehicular environments. This is further enhanced by leveraging cutting-edge 5G-based mmWave communication, which offers high-speed data transfer and low-latency communication, crucial for minimizing offloading delays in dense vehicular networks. The combination of the multi-hop offloading model and mmWave technology provides a distinct edge over related works, significantly improving system performance in terms of delay reduction and overall resource efficiency.

As depicted in all performance evaluations, the MVTO-PPO scheme not only surpasses traditional approaches like A3C and Random but also sets a new benchmark for efficiency in VEC systems. Its ability to adapt to dynamic vehicular environments and exploit advanced 5G-NR-V2X RAT further solidifies its role as a state-of-the-art solution for optimizing multi-hop task offloading. These distinctive characteristics position the MVTO-PPO scheme as a superior choice for addressing the latency and resource allocation challenges inherent in VEC scenarios, making it a robust and scalable solution for future vehicular networks.

7. Conclusion

In this paper, we proposed a task offloading scheme by utilizing multi-hop vehicle computation resources in Vehicle to Vehicle (V2V) communication based on mobility analysis. This approach involves identifying multi-hop vehicles that satisfy the necessary communication and computation requirements to execute tasks offloaded by the client vehicle and the single-hop vehicles present near the client vehicle. We formulate an NP-hard optimization problem for task offloading to minimize all tasks' weighted sum of computation delay. For this, a proximal policy optimization-based Multi-hop Vehicular Task Offloading (MVTO-PPO) scheme in vehicular edge computing is designed for low complexity that provides the optimal solution. We abstracted the task offloading process as a Markov decision process. A deep reinforcement learning-based algorithm is designed to make decisions about



(a) Varying No. of Hops (b) Considering 4-Hops

Fig. 8. Delay analysis of algorithms with varying No. of multi-hops.

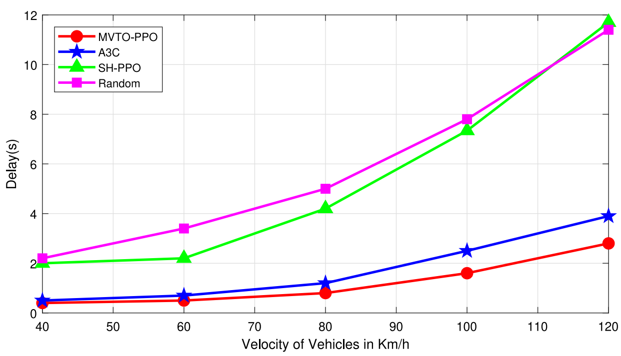


Fig. 9. Delay analysis while varying vehicular velocity.

offloading tasks to vehicles to achieve long-term benefits by improving environmental perception quality through appropriate task offloading. To enhance system performance by exploring the fifth-generation new-radio vehicle-to-everything communication model, which includes cellular connection and millimeter wave. By comparing it to benchmark techniques, the simulation results show that the proposed algorithm can effectively decrease the offloading delay of tasks. In future research, we plan to extend the model with 6G technologies, analyze the impact of vehicular density and mobility on performance, and refine reward functions for multi-agent reinforcement learning. We will explore AI-based resource management and dynamic pricing to incentivize resource sharing.

CRediT authorship contribution statement

Manzoor Ahmed: Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Salman Raza:** Visualization, Validation, Software, Project administration, Methodology, Formal analysis, Data curation. **Haseeb Ahmad:** Writing – review & editing, Project administration, Methodology, Investigation. **Wali Ullah Khan:** Writing – review & editing, Supervision, Resources, Formal analysis, Data curation. **Fang Xu:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Khaled Rabie:** Writing – review & editing, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Raza, S. Wang, M. Ahmed, M.R. Anwar, M.A. Mirza, W.U. Khan, Task offloading and resource allocation for IoV using 5G NR-V2X communication, IEEE Internet Things J. 9 (13) (2022) 10397–10410.
[2] K. Yu, L. Lin, M. Alazab, L. Tan, B. Gu, Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4337–4347.
[3] L. Zhao, W. Zhao, A. Hawbani, A.Y. Al-Dubai, G. Min, A.Y. Zomaya, C. Gong, Novel online sequential learning-based adaptive routing for edge software-defined vehicular networks, IEEE Trans. Wireless Commun. 20 (5) (2021) 2991–3004.
[4] X. Huang, S. Leng, S. Maharjan, Y. Zhang, Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks, IEEE Trans. Veh. Technol. 70 (9) (2021) 9282–9293.
[5] K. Zhang, J. Cao, Y. Zhang, Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks, IEEE Trans. Ind. Inform. 18 (2) (2022) 1405–1413.
[6] Y. Cui, D. Zhang, T. Zhang, L. Chen, M. Piao, H. Zhu, Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices, AEU-Int. J. Electron. Commun. 118 (2020) 153134.
[7] M. Ahmed, S. Raza, M.A. Mirza, A. Aziz, M.A. Khan, W.U. Khan, J. Li, Z. Han, A survey on vehicular task offloading: Classification, issues, and challenges, J. King Saud Univ.-Comput. Inf. Sci. 34 (7) (2022) 4135–4162.
[8] S. Raza, S. Wang, M. Ahmed, M.R. Anwar, A survey on vehicular edge computing: architecture, applications, technical issues, and future directions, Wirel. Commun. Mob. Comput. 2019 (1) (2019) 3159762.
[9] Z. Degan, W. Shuo, Z. Jie, Z. Haoli, Z. Ting, Z. Xiumei, A content distribution method of internet of vehicles based on edge cache and immune cloning strategy, Ad Hoc Netw. 138 (2023) 103012.
[10] J. Feng, Z. Liu, C. Wu, Y. Ji, Mobile edge computing for the internet of vehicles: Offloading framework and job scheduling, IEEE Veh. Technol. Mag. 14 (1) (2019) 28–36.
[11] K. Zhang, Y. Mao, S. Leng, Y. He, Y. Zhang, Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading, IEEE Veh. Technol. Mag. 12 (2) (2017) 36–44.
[12] S. Raza, M.A. Mirza, S. Ahmad, M. Asif, M.B. Rasheed, Y. Ghadi, A vehicle to vehicle relay-based task offloading scheme in vehicular communication networks, PeerJ Comput. Sci. 7 (2021) e486.
[13] D.-g. Zhang, Y.-y. Cui, T. Zhang, New quantum-genetic based OLSR protocol (QG-OLSR) for mobile ad hoc network, Appl. Soft Comput. 80 (2019) 285–296.
[14] S. Raza, W. Liu, M. Ahmed, M.R. Anwar, M.A. Mirza, Q. Sun, S. Wang, An efficient task offloading scheme in vehicular edge computing, J. Cloud Comput. 9 (1) (2020) 1–14.
[15] L. Chen, D.-G. Zhang, J. Zhang, T. Zhang, W.-J. Wang, Y.-H. Cao, A novel offloading approach of IoT user perception task based on quantum behavior particle swarm optimization, Future Gener. Comput. Syst. 141 (2023) 577–594.
[16] D. Zhang, G. Li, K. Zheng, X. Ming, Z.-H. Pan, An energy-balanced routing method based on forward-aware factor for wireless sensor networks, IEEE Trans. Ind. Inform. 10 (1) (2014) 766–773.
[17] I. Sorkhoh, D. Ebrahimi, R. Atallah, C. Assi, Workload scheduling in vehicular networks with edge cloud capabilities, IEEE Trans. Veh. Technol. 68 (9) (2019) 8472–8486.
[18] J. Du, F.R. Yu, X. Chu, J. Feng, G. Lu, Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization, IEEE Trans. Veh. Technol. 68 (2) (2019) 1079–1092.

[19] X. Zhang, M. Peng, S. Yan, Y. Sun, Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications, IEEE Internet Things J. 7 (7) (2019) 6380–6391.

[20] Y. He, N. Zhao, H. Yin, Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach, IEEE Trans. Veh. Technol. 67 (1) (2018) 44–55.

[21] C. Chen, L. Chen, L. Liu, S. He, X. Yuan, D. Lan, Z. Chen, Delay-optimized v2v-based computation offloading in urban vehicular edge computing and networks, IEEE Access 8 (2020) 18863–18873.

[22] J. Zhang, M.-j. Piao, D.-g. Zhang, T. Zhang, W.-m. Dong, An approach of multi-objective computing task offloading scheduling based NSGS for IOV in 5G, Cluster Comput. 25 (6) (2022) 4203–4219.

[23] D.-G. Zhang, W.-M. Dong, T. Zhang, J. Zhang, P. Zhang, G.-X. Sun, Y.-H. Cao, New computing tasks offloading method for MEC based on prospect theory framework, IEEE Trans. Comput. Soc. Syst. 11 (1) (2024) 770–781.

[24] D.-G. Zhang, C.-H. Ni, J. Zhang, T. Zhang, Z.-H. Zhang, New method of vehicle cooperative communication based on fuzzy logic and signaling game strategy, Future Gener. Comput. Syst. 142 (2023) 131–149.

[25] D.-G. Zhang, J.-X. Wang, J. Zhang, T. Zhang, C. Yang, K.-W. Jiang, A new method of fuzzy multicriteria routing in vehicle ad hoc network, IEEE Trans. Comput. Soc. Syst. 10 (6) (2023) 3181–3193.

[26] D.-g. Zhang, J. Zhang, C.-h. Ni, T. Zhang, P.-z. Zhao, W.-m. Dong, New method of edge computing based data adaptive return in internet of vehicles, IEEE Trans. Ind. Inform. 20 (2) (2024) 2042–2052.

[27] D. Zhang, H. Ge, T. Zhang, Y.-Y. Cui, X. Liu, G. Mao, New multi-hop clustering algorithm for vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 20 (4) (2019) 1517–1530.

[28] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, L. Gui, X. Shen, Cooperative task scheduling for computation offloading in vehicular cloud, IEEE Trans. Veh. Technol. 67 (11) (2018) 11049–11061.

[29] H. Wang, X. Li, H. Ji, H. Zhang, Federated offloading scheme to minimize latency in MEC-enabled vehicular networks, in: 2018 IEEE Globecom Workshops, GC Wkshps, IEEE, 2018, pp. 1–6.

[30] C. Chen, Y. Zeng, H. Li, Y. Liu, S. Wan, A multihop task offloading decision model in mec-enabled internet of vehicles, IEEE Internet Things J. 10 (4) (2023) 3215–3230.

[31] D. Zhang, L. Cao, H. Zhu, T. Zhang, J. Du, K. Jiang, Task offloading method of edge computing in internet of vehicles based on deep reinforcement learning, Cluster Comput. 25 (2) (2022) 1175–1187.

[32] K. Xiong, S. Leng, C. Huang, C. Yuen, Y.L. Guan, Intelligent task offloading for heterogeneous V2X communications, IEEE Trans. Intell. Transp. Syst. 22 (4) (2021) 2226–2238.

[33] S. Zhou, Y. Sun, Z. Jiang, Z. Niu, Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks, IEEE Commun. Mag. 57 (5) (2019) 49–55.

[34] X. Wang, Z. Ning, S. Guo, L. Wang, Imitation learning enabled task scheduling for online vehicular edge computing, IEEE Trans. Mob. Comput. 21 (2) (2022) 598–611.

[35] D. Zhang, W. Wang, J. Zhang, T. Zhang, J. Du, C. Yang, Novel edge caching approach based on multi-agent deep reinforcement learning for internet of vehicles, IEEE Trans. Intell. Transp. Syst. 24 (8) (2023) 8324–8338.

[36] L. Chen, D.-g. Zhang, J. Zhang, T. Zhang, J.-y. Du, H.-r. Fan, An approach of flow compensation incentive based on Q-learning strategy for IoT user privacy protection, AEU-Int. J. Electron. Commun. 148 (2022) 154172.

[37] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, A. Ylä-Jääski, Folo: Latency and quality optimized task allocation in vehicular fog computing, IEEE Internet Things J. 6 (3) (2019) 4150–4161.

[38] S. Raza, M. Ahmed, H. Ahmad, M.A. Mirza, M.A. Habib, S. Wang, Task offloading in mmWave based 5G vehicular cloud computing, J. Ambient Intell. Humaniz. Comput. 14 (9) (2023) 12595–12607.

[39] Z. Li, L. Xiang, X. Ge, G. Mao, H.-C. Chao, Latency and reliability of mmWave multi-hop V2V communications under relay selections, IEEE Trans. Veh. Technol. 69 (9) (2020) 9807–9821.

[40] 3GPP, Study on enhancement of 3GPP support for 5G V2X services (v16.2.0, release 16), 2019, 3gpp rel 16, no. TR 22.886.

[41] J. Zou, T. Hao, C. Yu, H. Jin, A3C-DO: A regional resource scheduling framework based on deep reinforcement learning in edge scenario, IEEE Trans. Comput. 70 (2) (2021) 228–239.