



Artificial Intelligence Inspired Task Offloading and Resource Orchestration in Intelligent Transportation Systems

Oshin Rawlley¹ · Shashank Gupta¹ · Jyotsana Chandrakar¹ · Manisha K. Johnson¹ · Chahat Kalra¹

Received: 12 December 2023 / Accepted: 27 August 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Internet of Vehicles (IoV) applications require the support of communication, caching, and computation (3C) resources to offload the computation-intensive tasks and for uplifting the traffic conditions in the development of sustainable smart cities. Intelligent Transportation Systems (ITS) lack the integrated ecosystems of addressing the low-latency task handovers, resource management issues, and centralized incentivization strategies. Digital Twin (DT) aids in capturing the real-time varying resource needs of the vehicles and the communication infrastructure that will regulate the task offloading process and facilitates in incentivizing the vehicular instances. In this manuscript, we establish a digital twin counterpart ($DT_{P\text{IoV}}$) of the physical IoV ($P\text{IoV}$) to meet the QoS requirements during dynamic offloading and the time-varying resource supply–demand of computationally intensive applications. We formulate a response delay minimization function which is solved by the proposed DT-driven context-aware dynamic offloading method (CADOM). Furthermore, we use M/M/1/N/FCFS queueing method that combats the drawbacks of handling the simultaneous deadline-based tasks in a volatile environment of $P\text{IoV}$. In addition, we also maximize the utilities of vehicle and RSU service satisfaction by employing a reward-based mechanism for on-demand allocation of resources based on the Stackelberg game, where the DT of vehicle is deemed as a leader and service provider RSUs as a follower. The simulation results establish that the proposed system outpaces the conventional traffic management system by emphasizing the role of $DT_{P\text{IoV}}$ in jointly optimizing the overall response latency for different task sizes and also ensure a better utility satisfaction by catering on-demand resource allocation.

Keywords Smart city transportation · Artificial intelligence · Digital twin · Task offloading · Resource allocation · Edge collaboration · Rewards · Internet of Vehicles (IoV)

Introduction

The realization of future smart cities development is facilitated due to the latest advancements in the scalable networks, communication technologies, and computational resource infrastructure [1–3]. The smart IoT devices in smart cities can be connected to the vehicles and communicate using wireless technologies for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications in physical internet of vehicles (PIoV) [4, 5]. The PIoV has also rendered various potentials with the digital twin (DT) technology and communication paradigms. To the extent of our current

understanding, the state-of-the-art has not studied the integration of low-latency task handovers and resource management [6, 7]. Though many authors have presented the PIoV architectures as layered frameworks [8–12] to cater the needs of other integrated environments. Figure 1 shows the development of adaptive IoV to establish a sustainable smart city infrastructure for processing the PIoV computation-intensive applications. The smart city facilitates dynamic interconnections supporting V2X communications, i.e., V2V and V2I [13]. We define the PIoV as a network of vehicles, equipped with limited computing and storage resources, allowed for local processing. The edge layer in PIoV comprises EIS-assisted RSUs placed on different intersections of the road, along with a base station located at the edge of the city. The EIS-assisted RSUs have a designated signal coverage area for catering the service requests. The EIS-assisted RSUs also effectively divide the road into distinct, non-overlapping segments, and the users within the PIoV network are distributed

✉ Shashank Gupta
shashank.gupta@pilani.bits-pilani.ac.in

¹ Department of Computer Science & Information Systems,
Birla Institute of Technology and Science, Pilani, Rajasthan,
India

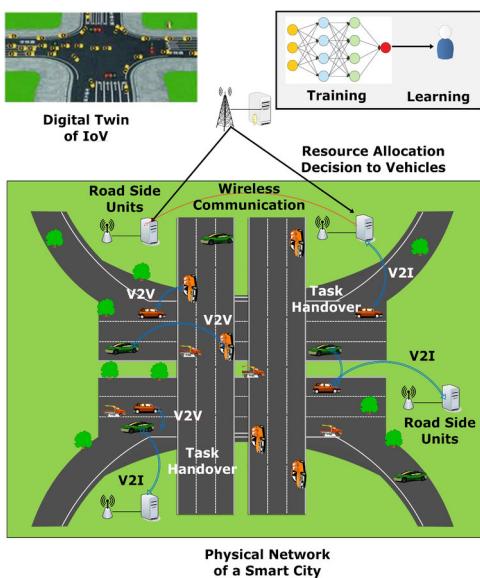


Fig. 1 Establishment of a sustainable smart city environment for deploying adaptive internet of vehicles (IoV)

across these segmented road sections. The DT gathers the real-time *PIoV* information via inter-twin communications [14], which includes physical scenario details such as the car's driving path, information about channel, and the service status of different computing units. Currently, research on DT-assisted IoV is in its growing phase. Few researchers [15, 16] in the related domain have made remarkable contributions in advancing the fusion of DT and IoV technologies. For instance, the study [17] presents a multiple-user DT-enabled IoV system for offloading that incorporates an adaptive service offloading (SOL) technique aimed at enhancing the QoS for users.

We intend to focus on optimizing network latency during task offloading using DT technology to facilitate smooth functioning of latency-critical applications [2]. In addition, we also capture the fluctuating resource demands of the vehicles and aim to maximize the vehicular service satisfaction in the dynamic environment of *PIoV*. This involves orchestrating the 3C (computing, communication, and caching) resources distribution through the collaboration between Edge Servers (ESs) in *PIoV*. The main objective for this cooperation is to reduce the network latency. Since the *PIoV*, i.e., the vehicles, roadside units (RSUs), and the digital -twin IoV (DT_{PIoV}) possess different utilities, the vehicle tries to capitalize on its service satisfaction which calculates the cumulative satisfaction derived from the RSUs. In the current scenario, AVs are resource-constrained and are incapable of fulfilling the diverse demands of computation-intensive Advanced Driver Assistance Systems (ADAS) applications [18–20, 20]. Some examples of such applications include

(but not limited to) parking assistance, lane change/keeping assistance, adaptive cruise control, self-driving assistance, collision avoidance, etc. AVs are mostly equipped with onboard units (OBUs) for performing small computations in the time-varying environment of IoV [21, 22]. However, the complex processing requirements in IoV still poses a huge challenge of how the incoming offloading tasks should be scheduled that have the same completion deadline. On the other hand, the advent of Roadside Units (RSUs), usage of LIDARs and radars, radio transceivers, etc., have made the brand-new technological frontier of IoV realizable. Nevertheless, the emerging computing service demands of such transport applications in the highly mutable environment of IoV also require a low-latency processing and deadline-based response [23–25].

Indeed, the *PIoV* has rendered various potentials with the DT and communication infrastructure; however, they face certain challenges such as low network latency for the high mobility applications owing to its restricted spatial range, stochasticity of the wireless channels, etc., [26, 27]. Moreover, the constrained deployment and operability of RSUs limits the ES computing capacity for neural network training. Here, DT aids in accurate predictions and monitoring *PIoV*, thereby enhancing system productivity and efficiency. It is also important to note that the complete dependence on only one type of network also causes the delay during the task offloading process [28]. If all the vehicular users access only one type of network to offload their tasks, then it will cause the issue of high latency [29]. Further, different vehicular users may have varied computational preferences [30] for quality of channel, tolerable transmission delay, computational resources requirement, etc. Therefore, it is necessary to have a dynamic resource orchestration to meet the diverse needs of the vehicular users. A dynamic DT should monitor the resource management, satisfaction levels of the vehicles, and RSUs. We formulate the vehicle satisfaction utility and reward-based resource management to reflect dynamic *PIoV* operations [31–35].

The existing research [36–38] have assumed predictable vehicle mobility patterns or data traffic behavior within specific time frames, neglecting the dynamic nature of the resource supply–demand in *PIoV* [39]. The centralized resource allocation (RA) schemes also contribute to be a bottleneck [40]. To facilitate the willingness of these networks, rewards are given as compensation to cover their service costs. For instance, when vehicles delegate their computation-intensive tasks to the nearby RSUs, it is essential to devise an appropriate incentive mechanism to encourage the RSUs for giving better services. However, the incentives and RA mechanism itself are computationally demanding and impose burden on the limited resources

of AV [41, 42]. In a nutshell, we conclude that there still lurk some technical challenges which we have addressed in this manuscript and are listed as follows: (a) How a minimal response time can be achieved during the task offloading process to the nearest available EIS. (b) How an overloaded EIS will schedule the simultaneous requests. (c) How to effectively incentivize the network resource to provide uninterrupted services to the vehicles.

Inspired by the above discussion, it is highly desirable to develop a DT-assisted task offloading method and unified resource management technique for efficient operational *PiOv* environment. We introduce a context-aware dynamic offloading method (CADOM) to capture the dynamic task offloading requests and fulfill them in low-response time. In addition, to maximize the QoS of the vehicles and RSUs, we also introduce a DT-driven reward-based mechanism that optimizes the resource supply–demand process in the *PiOv* environment. The manuscript lists its main contributions as follows:

- To ensure low-latency processing in DT_{PiOv} during task offloading, we propose a novel context-aware dynamic offloading method (CADOM). Unlike the existing task offloading techniques [39, 40], CADOM dynamically selects the available EIS based on monitoring their availability status (active/idle).
- To effectively incentivize the vehicles and EIS-assisted RSUs, we develop a two-phase reward-based mechanism based on the Stackelberg game.
- We formulate a multi-optimization function for minimizing the global latency of the task offloading request and also a service maximization function for vehicles and RSUs, thereby facilitating optimization of the 3C resources through the different EIS cooperation.
- Finally, we conduct our comprehensive simulations to prove the effectiveness and relevance of incorporating DT_{PiOv} in catering to the high response times of the computation-intensive task.

The structure of the manuscript is organized into nine sections. We study the literature art in “[Related State-of-the-Art](#)” section. “[System Model](#)” section introduces the system model of the proposed *PiOv* and its corresponding DT_{PiOv} framework. “[Metrics of Performance Analysis](#)” section shows the different metrics for analyzing the performance of the proposed DT_{PiOv} scenario. “[Vehicular Satisfaction Index](#)” section discusses the vehicular satisfaction index followed by “[Problem Formulation](#)” section of problem formulation to solve the optimization functions. “[Algorithms](#)” section gives the algorithms designed followed by the performance analysis in “[Performance Analysis](#)” section, where the simulation procedure is elaborated. Finally, “[Conclusion](#)” section concludes this article.

Related State-of-the-Art

Numerous contemporary works have been carried out for autonomous vehicles to develop a smart community and smart sustainable ITS. This section firstly presents existing works that jointly study the task offloading strategies, followed by the latest advancements in resource allocation methods in *IoV*. We also discuss the role of DT in *IoV* and, lastly, the existing works on incentivizing the resources for encouraging them in providing better services to the vehicular users in a smart city scenario.

Task Offloading Mechanisms

Computation offloading defines the offloading of the computation-intensive tasks to an EIS having higher computational capabilities. The existing works focus on mitigating high latency in the offloading process, as these computation-intensive tasks are delay-sensitive. In hybrid offloading, the mode of task offloading depends highly on the nature of the task. Tasks that require minimal computational power but involve significant data can be executed locally. On the other hand, tasks that demand substantial computation but involve limited data transmission can be offloaded to the EIS, remote cloud, or other available vehicles [16]. Consequently, when a vehicle is responsible for handling various categories of tasks, a combination of multiple offloading strategies can be simultaneously considered. In the recent work [43], tasks are categorized into three sections, where one section involves local computing, second section involves offloading of tasks to other vehicles taking n hops leveraging V2V communication, and the third section comes into picture when the task gets offloaded to the EIS for computation. Y. Wang et al. [44] introduced an algorithm called VCMO (V2I/V2V hybrid multi-hop offloading), aiming to minimize the offloading overhead in the *IoV* environment. However, this work primarily focuses on addressing the issue of computation-intensive task offloading overhead and does not focus on the candidate vehicle selection and scheduling part in multi-hop offloading.

Further, there has been growing interest in solving the joint task offloading and efficient RA issues in *IoV* networks. For the task offloading scenario, many *IoV* applications, such as emergency braking and safe driving assistance, require low latency. To address this, researchers have focused on multi-access edge computing (MEC)-equipped *IoV*, where EISs are identified for offloading and allocating computing resources to the vehicles for reducing the vehicle’s high response latency. For instance, [45] proposed an EC approach for parking that utilizes parked vehicles to assist EISs in executing the offloaded tasks. They designed a task scheduling algorithm along with the techniques for selecting EIS and managing resources to enhance the performance of

offloading. Similarly, Wang et al. [43] presented an ad hoc vehicular fog network where the vehicles in vicinity act as federated network nodes. The authors proposed a vehicular EC-empowered policy to minimize the overall task execution time. Such works demonstrate that collaborating EC is an impactful method to enhance IoV system performance by reducing delays of system responses.

For RA issues, the ineffective network resource distribution and vehicular requests varying across different topographical regions pose a resource imbalance problem. In addition, owing to the resource insufficiency in the IoV environment during the process of executing the computation-intensive tasks, there is necessity of optimal resource allocation scheme. The EISs installed in hotspot areas often experience high utilization, while those in remote locations suffer underutilization. Therefore, effective RA strategies are needed to achieve the load balancing in cooperative MEC scenarios [46, 47]. In a study referenced as [48], a model for managing the edge caching and computing is presented, which enhances the allocation of resources, service caching, and scheduling approaches. The scheme has been shown through the theoretical analysis and simulation results to achieve the delay having less optimal performance while adhering to time-stricken budget constraints. Another notable contribution, referenced as [49, 50], focuses on enhancing the caching hit rate. They address the challenges related to content caching and determining the optimal cache size. Furthermore, Saleem et al. [51] addressed the interdependence between mobility-offloading framework and performance in MEC networks. They developed a strategy for allocating the tasks and assigning some power policy that comprehensively considers the user mobility, characteristics of task, decentralized resources, and the limitations over energy of the mobile devices.

Additionally, an AI-driven collaborative RA system is suggested in [52] that improves not only the utilization of resources at an average rate but also decreases the average decision delay time. Another study [53] focuses on resolving the content caching issue, where the authors proposed Deep Deterministic Policy Gradient (DDPG) which is a nature-inspired method to enhance the content hit rate and decrease the latency for content delivery. We also explored the recent works that studied both issues: computation task offloading and resource allocation jointly. Alfakih et al. [54] introduced a State-Action-Reward-State-Action (RL-SARSA) algorithm which is based on Reinforcement-Learning to tackle the resource management challenge in EIS. The algorithm aims to make optimal offloading decisions that minimize overall system costs, considering the energy consumption and delays in computing time. On the other hand, Xu et al. [55] developed Fuzzy-Task-Offloading-and-Resource-Allocation (F-TORA), based on Takagi-Sugeno Fuzzy Neural Network (T-S FNN) and game

theory. Meanwhile, other studies focus on joint offloading and hybrid resource allocation, considering both computing and communication resources. For instance, Zhou et al. [56] proposed the Joint Offloading Proportion and Resource Allocation Optimization (JOPRAO) algorithm with the goal of reducing the completion time of the task.

DT-Driven IoV

According to [57], DT can be utilized for constructing the digital replica where the real-time operational condition of the physical system is examined. DT presents a virtual representation of an IoV system that undergoes continuous updates with the data related to its maintenance and performance throughout its lifespan. The DT environment can accurately depict the real-world traffic situations and produce representative traffic information by fine-tuning the parameters and modeling a vast volume of actual traffic information taken over an extended period. Therefore, it serves as a foundation for making informed decisions regarding traffic dispatch. Within the DT space, several stages of virtual simulation models can be created, facilitating tasks such as task offloading and RA. These tasks offer valuable guidance for practical decision-making processes. Additionally, the enormous and extensive data stored in the DT Data Center (DTDC) can be leveraged to train the AI algorithms, ensuring the accuracy of predictions. In the domain of AVs, real traffic information serves as the foundation for creating high-fidelity 3D virtual environments that can accurately replicate the actual traffic scenes. By utilizing advanced capabilities such as high-end maps (OSM, HERE, etc.) and leveraging game engines (UNITY) to simulate diverse weather conditions, the DT-assisted IoV world can generate an extensive range of test scenarios derived from the real data. This complete support for AV driving aids thorough evaluations and validations of AVs in various realistic driving conditions.

In the recent times, some eminent researchers started utilizing DT expertise in the field of IoV for enhancing capabilities of ITS. Xu et al. [17] aimed to enhance the QoS in terms of feedback time within a multi-user offloading scenario in the context of DT-enabled IoV. They proposed a SOL method utilizing deep reinforcement learning to achieve the efficient offloading approach. Results of simulation built on actual operational facts in Nanjing validated the effectiveness and adaptability of the SOL scheme. In another study, Liu et al. [58] focused on capturing the driving state of heavy vehicles. They developed two models of DT using Gaussian process and deep CNN techniques individually. The application scope of these models was studied, and further scope of research in this domain was predicted. The researchers envisioned that the MEC technology, along with game engines, 3D reconstruction, and other cutting-edge technologies, can easily support the DT-IoV system.

Within the DT-IoV framework, traffic administrators have the flexibility to manipulate timelines, space, and applicative scenarios based on real-time information, physical rules of traffic, and their functioning logic. DT-assisted IoV offers an exceptional platform for effective cost management and efficient reduction of future transportation developments [59]. Overall, DT has been applied in other related studies for assisting drivers [60] and enhancing safety in driving [61]. Additionally, DT is utilized to facilitate task offloading with the help of MEC environment. In [39], the authors introduced a mobile offloading strategy that considers the impact of user mobility and the unpredictable nature of the MEC environment. The proposed scheme aims to minimize the offloading time while taking into account the accumulated costs of service migration. It also addresses the discrepancy between the DT space and the physical world of IoV system.

Incentive-Based Resource Management Schemes

Incentive-based mechanisms in resource management schemes are preferred to encourage the network infrastructure in the IoV system to provide their 3C resources for vehicles. For, e.g., when vehicles offload their computation-intensive tasks to the neighboring EIS, satisfactory rewards are essential for RSUs to provide uninterrupted facilities back to vehicles. Zhou et al. [62] introduced an effective contract theoretical modeling-based incentive mechanism, which optimizes the probable usefulness of the base station taking into consideration the exceptional features of every type of vehicle. On the other hand, for the purpose of edge caching, Zheng et al. [63] devised a scalable and unified Stackelberg game that divides the game into two parts: user and storage allocation games. A distributed approach was utilized by the Stackelberg game-based ADMMs (Alternating Direction Method of Multipliers) to solve these games. Wang et al. [64] addressed the issues related to computational offloading, allocation of resources, and schemes related to caching of content as optimization problems. They designed a procedure based on ADMM to solve these optimization problems. In the pursuit of maximizing usefulness of movable virtual network operators, Liang and Yu [65] proposed a RA method and expressed it as an optimization problem. Additionally, they established a competent distributed ADMM algorithm which virtually allocates the resources for virtual wireless networks. However, these approaches are mainly tailored for static networks and, hence, are incapable to be utilized in highly dynamic and unpredictable network conditions of the IoV. However, most of the recent techniques assume mobility patterns to be foreseeable which is unrealistic in the real-time environment of IoV during the task offloading process. Note that most of the existing works for the task offloading schemes exceed the tolerable

latency of the task completion and do not consider the real-time deviations when DT technology is utilized. In addition, the incentive-based resource allocation strategies are mostly centralized and also overburden the computational resources, thereby failing to cater to the time-varying demands of the vehicular users. Keeping in mind the above discussion, there is a requisite of joint consideration of computational task offloading problem and incentive-based dynamic resource allocation simultaneously using DT technology. Hence, in this paper, we exclusively utilize the DT technology to solve both the time-varying computing problems as well as manage the resource supply–demand based on reward-based resource orchestration [66].

System Model

To address the conjoint process of task offloading and RA, we first elaborate the service architecture of the physical IoV (*PIoV*) by describing the scenario and constructing its equivalent digital twin model (DT_{PIoV}). We also model the V2I communication with three different types of transmission modes (local, edge, and cloud) to enable the cooperative handshake among the different EIS-assisted RSUs for task offloading in *PIoV* structure. Furthermore, we also discuss the performance metrics of the described scenario and conduct a performance analysis in the simulation section. For the RA, we first discuss the maximization of the utilities of the vehicles using the Stackelberg game [67]. Finally, we formulate two optimization problems to solve (1) the response delay minimization function in CADOM under some well-defined assumptions and real-time constraints and (2) the utility maximization of vehicles and EIS-assisted RSUs using a reward-based mechanism. Tables 1 and 2 depict the main notations and abbreviations used in our work.

Scenario Description

We consider an urban scenario of vehicular nodes that consists of multiple EIS-assisted RSUs and one cloud-assisted main base station for establishing a low-latency edge-oriented V2X offloading model as shown in Fig. 2. This model would facilitate the overloaded resource-constraint vehicles to offload the computation-intensive tasks to the other edge-assisted resources. We assume a set of all EIS-assisted RSUs as static edge nodes which are denoted as $SEN = SEN_1, SEN_2, \dots, SEN_n$. Let I be the set of vehicles that are considered mobile edge nodes denoted as, $I = \{MEN_1, MEN_2, \dots, MEN_i\}$. All these MEN_i are grouped in different clusters, say C_1, C_2, \dots, C_n . MEN_i generates $j = \{1, 2, \dots, j\}$ tasks that are offloaded to two different tiers, i.e., edge tier and cloud tier. Further, every MEN_i is

Table 1 Notational meanings

Notation	Meaning
DT_{PIoV}	Digital twin of PIoV structure
PE_{PIoV}	Physical space of entities in DT_{PIoV}
V_{PIoV}	Virtual entities
S_{PIoV}	Services provided in the DT_{PIoV}
D_{PIoV}	Data in DT_{PIoV}
$COM_{PS \rightarrow DS}$	Communication channel information between physical space to digital space
C_l	Local computing capability
S_l	Local storage capability
C_E	Edge computing capability
S_E	Edge storage capability
r_d	Coverage range with radius
$T(MEN_i, j)$	Computation intensive task
$D_{MEN,j}^s$	Input data size
$\partial_{MEN, j}$	Computational requirement of the task
$\tau_{MEN_i, j}^l$	Computation capacity of local server
$t_{MEN, j}^{max}$	Maximum tolerable latency for task completion
d_o	Binary decision variable for offloading decision
$t_{mode_{i,j}}$	Transmission mode
γ_i^{mode}	Signal to interference plus noise ratio
p_d	Total processing delay
$Real_{trans}^{MEN_i \rightarrow SEN_k}(t)$	Transmission time taken by MENs to offload data to static edge-node SEN_k
s_j	Binary variable for indicating the state of the DT
$T_{forw}^{SEN_k \rightarrow SEN_j}(t)$	Forwarding time from static edge-node k to j
$T_{exe}^{SEN_j}(j)$	Execution time
$T_{feed}^{SEN_j \rightarrow i}(t)$	Feedback time
$T_{queue}^{SEN_k}(j)$	Queueing time
$\tau_{MEN_i, j}^{eis,n}$	Computation capacity of the EIS j
$p_{i,n}$	Preference of the vehicle i to SEN n
r_n^{idle}	Accessible computational resource
W_n	Cumulative preference of all MENs for all SEN_n
C_n	Cost per CPU frequency unit
p^*	Resource allocation policy

equipped by local computing and storage capabilities, say, C_l and S_l , for local data analysis. Moreover, SEN_n have computing and storage capabilities C_E and S_E , respectively, to provide the services over the EIS-assisted RSU coverage range of radius r_d to the vehicles. The computing and storage constraints for the PIoV scenario are as follows: $C_E \gg C_l$, $S_E \gg S_l$. We also assume that one main base station (MBS) is optimally placed for maximum geographic coverage to

Table 2 Main abbreviations

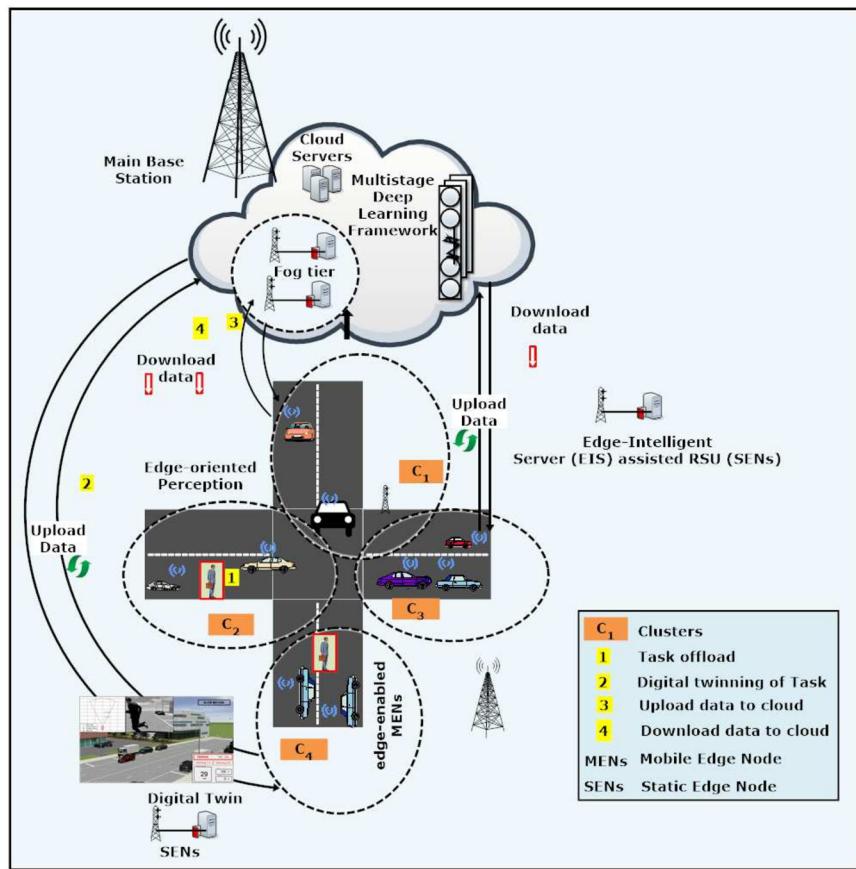
Abbreviation	Explanation
IoV	Internet of Vehicles
OBUs	Onboard Units
DT	Digital Twin
5G	Fifth-Generation
ITS	Intelligent Transportation Systems
AV	Autonomous Vehicles
ADAS	Advanced Driver Assistant Systems
EC	Edge Computing
V2X	Vehicle to Everything
MBS	Main Base Stations
RSUs	Road Side Units
EI	Edge-Intelligent
RA	Resource Allocation
EIS	Edge-Intelligent Servers
URLLC	Ultra-reliable Low Latency Communications
PloV	Physical Scenario of Internet of Vehicles
MEC	Multi-Access Edge Computing
PE	Physical Entities
DE	Digital Entities
SEN	Static Edge Nodes
MEN	Mobile Edge nodes
SINR	Signal to interference plus noise ratio
CADOM	Context-aware dynamic offloading method
VSI	Vehicular satisfaction index
NE	Nash Equilibrium

orchestrate the EIS-assisted RSUs and conduct an efficient management of the network resources.

Digital Twin of PIoV Scenario (DT_{PIoV})

We render the PIoV scenario digitally by constructing a DT-empowered physical IoV system DT_{PIoV} . A digital twin is a virtual model representing all the vehicular entities depicted in the PIoV scenario such as EIS-assisted RSUs, vehicles, and cloud-connected base stations. Corresponding to Fig. 2, Fig. 3 explains the DT_{PIoV} where the MENs request for the task offloading in the left part and the DT_{PIoV} services are elaborated in the digital space of the diagram depicted in the upper part of Fig. 3. The interactions among the components in the PIoV that are depicted in the middle portion of Fig. 3 are supported by digital simulations and software illustrated in the digital space above in Fig. 3. The purpose of the digital representation for the real-time PIoV entities is to facilitate seamless offloading of AI-enabled tasks on the road. The digital twin of the PIoV structure, i.e., DT_{PIoV} , is shown in the upper part of Fig. 3. To define the DT_{PIoV} entities, we express the digital representation of the PIoV system in

Fig. 2 Illustration of physical structure of IoV (PIoV) in sustainable smart city infrastructure



the following tuple below:

$$DT_{PIoV} = \{P_{PIoV}, V_{PIoV}, DA_{PIoV}, S_{PIoV}, COM_{PIoV}^{PS \rightarrow DS}\} \quad (1)$$

Here, P_{PIoV} represents the physical space of entities in DT_{PIoV} environment comprising vehicular nodes, i.e., MEN and SENs, EIS-assisted RSUs, and one cloud-assisted MBS. V_{PIoV} corresponds to the virtual entities that are obtained by virtually mapping the P_{PIoV} to the digital space and by simulating their semantics, behavior, rule models, etc. [48]. DA_{PIoV} is utilized for representing the data in DT_{PIoV} , which includes the data in PE_{PIoV} such as the speed of vehicular nodes and their operational environment. S_{PIoV} consists of numerous services provided in the DT_{PIoV} including real-time simulation, dynamic optimization, etc. $COM_{PIoV}^{PS \rightarrow DS}$ is also applied for representing the channel for communication between the vehicular node and EIS-assisted RSU in P_{PIoV} and the virtual communication link in V_{PIoV} . The MEN_i is considered to have heterogeneous multiaccess communication channel, and the information of the communication channels between physical space to digital space is represented using $COM_{PIoV}^{PS \rightarrow DS}$. In order to synthesize an analysis, we consider a few vehicles on the road to simulate our proposed scenario. The

idea is to scale up the synthesis to multiple vehicles on the road. The sensor-equipped MEN_i perceives the operational environment details DA_{PIoV} such as vehicle speed, GPS information, trajectories information at the intersections, and vehicle destination using radars, LIDARS, and GPS. The cloud-assisted MBS plays the role of network resource manager and data distributor in DT_{PIoV} . It also offers traditional MBS functions as well as stores massive DT data within its coverage area. S_{PIoV} are the system services that are provided in the DT_{PIoV} which conducts traffic simulations, modeling of the traffic entities, dynamic decision-making, scheduling services, and provides real-time traffic analytics. The DT_{PIoV} data can comprise of the $COM_{PIoV}^{PS \rightarrow DS}$ which represents the communication status between each P_{PIoV} and V_{PIoV} . In addition, to reach out to the cloud-assisted MBS, the vehicle also reaches out to the geographically dispersed EIS-assisted RSUs via a wireless channel. We establish the property of self-adaptive balance between network access mode (EIS-assisted RSU or cloud-assisted MBS) and offloading strategy. To facilitate reliable low-latency services, every RSU is equipped with an EIS having sufficient computing and storage capability. These EIS-assisted RSUs communicate through optical fibers. The information on the RSUs and EIS's current status such as busy/idle state is shared to construct V.

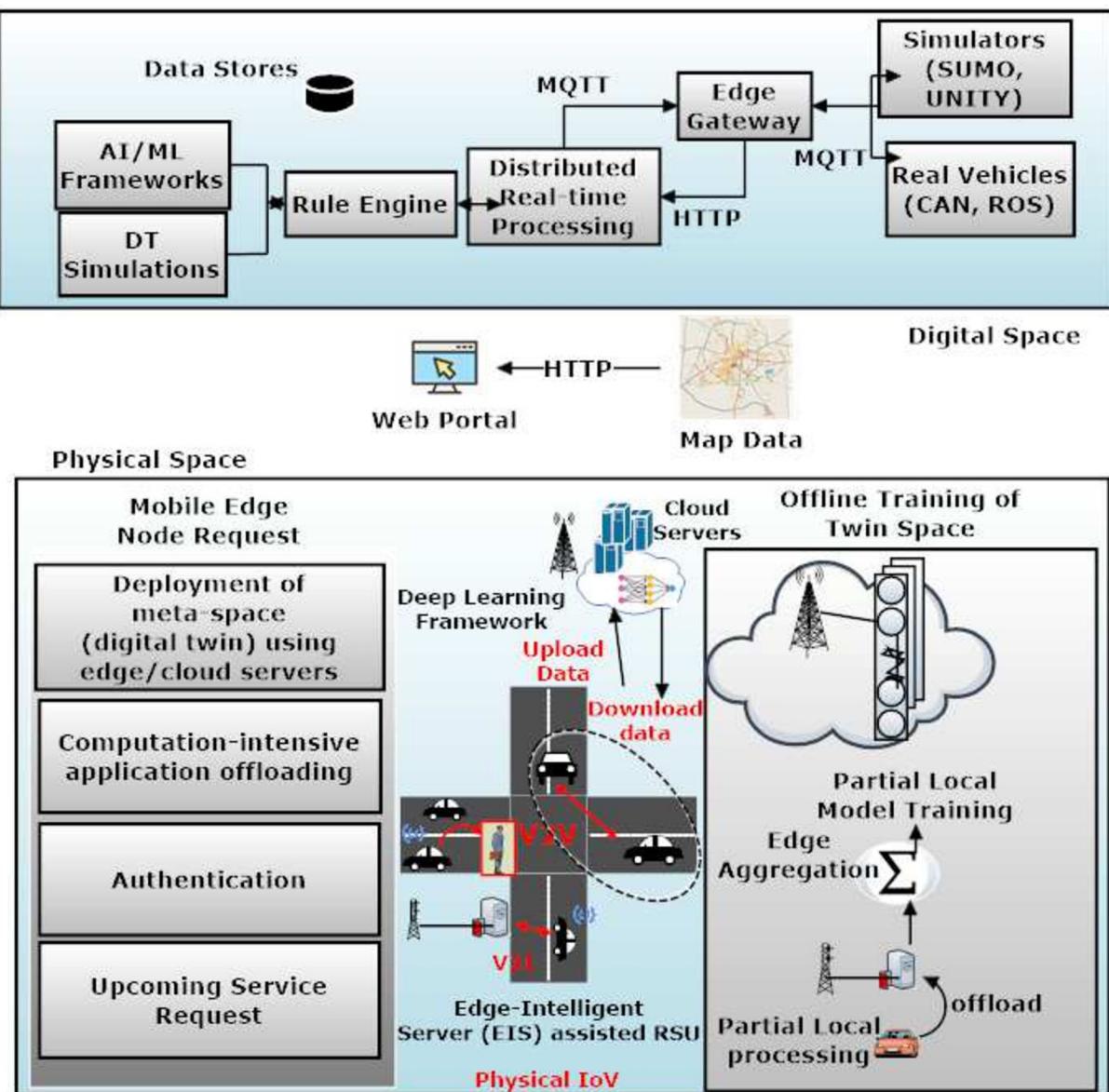


Fig. 3 Digital mapping of physical entities in the PIoV named as digital twin of PIoV (DT-PIoV)

In order to manage the computation-intensive tasks better, they are forwarded to SENs. In an assumption that the closest EIS-assisted RSU to the vehicle is busy, the vehicle request can further be forwarded to another available EIS-assisted RSU using the forwarding principle. We also implement the queue data structure to maintain the vehicular request in a buffer when an EIS-assisted RSU is already overloaded with simultaneous deadline-sensitive requests. Suppose an EIS-assisted RSU is supposed to serve five requests simultaneously at a time and if some more requests arrive then, the resource is said to be overloaded. Therefore, we utilize the queueing concept where the upcoming requests can be queued in a buffer, so that as soon as the initial five requests are served, the next buffered requests already line up can be

served simultaneously. The data stored in the cloud-assisted MBS is used for providing enhanced traffic analysis which the EIS-assisted RSUs cannot support due to their limited computing capabilities. Further, tasks such as offline training of AI algorithms, advanced modeling, and simulations can be conducted over the cloud-assisted MBS, which do not require immediate responses in real time. The reason for the offline training of AI algorithms gives an upper edge to establish a foundation for interpreting and analyzing the traffic scenarios. Since there is continuous change in the traffic conditions and vehicle mobility, the scenarios are bound to change. Therefore, periodic training of the pre-trained algorithms is required according to the varying changes in the IoV environment. The potential applications are object detect, cruise

control, lane merge etc., which require pre-training. In Fig. 3, the virtual spaces consisting of co-simulation platforms such as SUMO, MATLAB, and UNITY are continuously updated by the interactions happening in the $P\text{IoV}$ scenario. Further, we describe the cloud-assisted MBS by making a workstation as a server that can suffice the requirements of our proposed $P\text{IoV}$ and $DT_{P\text{IoV}}$ environments.

To analyze the $P\text{IoV}$ scenario, the traffic data is referred from TomTom [66]), and map data is leveraged from OpenStreetMap [67], all of which can be integrated into AWS API Gateway via HTTP. Incorporation of such facilities will empower the deployment of yet more authentic information about the traffic scenes in the $DT_{P\text{IoV}}$, thereby providing improved management and control towards SEN and MEN in the $P\text{IoV}$. Moreover, an interactive web portal is envisioned to visualize the workflow of the $DT_{P\text{IoV}}$ on the cloud and allow the end users to function well with the microservices. The position and speed data can be uploaded to AWS IoT Core via MQTT protocol. In addition, an edge gateway is deployed to permit external simulators such as Unity [70] and SUMO [69] to conduct the communications with AWS IoT Core through MQTT. We have used MQTT as we are dealing with resource-constrained environment and want to use minimum bandwidth. Moreover, the protocol has a lower overhead and can handle numerous devices essential for scaling in DT implementation. It also ensures reliable message delivery, thereby increasing the quality-of-service (QoS).

Context-Aware Dynamic Offloading Method (CADOM)

The proposed CADOM allows MEN to initialize the incoming computational task and transmit its details in the form of input data size, computational intensity (measured by the total CPU cycles needed for the task), and QoS requirements to the EIS-assisted RSUs. Further, more information is continuously updated by MENs and EIS-assisted RSUs, such as movement speed, location, and computational capability. The incorporation of both types of information aids in making the decision for offloading to cater to the QoS requirements.

The computing capability of the RSU is determined by the RAM and the memory storage provided in the ES. The RSUs are assisted by the ES for advanced computations required during the task offloading. However, the computational capacities of the RSUs are not enough to conduct high-end traffic analytics. The traffic administrators deduce many interpretations and insights based on the traffic conditions such as traffic flow forecasting, accident-prone areas, alarming weather conditions, and congestion intersections which require more processing and storage. This cannot be provided by the ES connected with RSUs. Therefore, we say that specific computing capabilities of EIS-assisted RSUs restrict certain tasks from being executed locally.

Moreover, to satisfy the QoS standards, the DT decides to offload the task to raise them to the appropriate hierarchy (edge/cloud). After receiving the task, EIS-assisted RSUs perform the task according to their available computational resources. In the existing state-of-the-art, [21]-[22] studied that the parked vehicles act as static communicating nodes for vehicles and the communication range in case of cloud computing paradigm limited the 9 distances of communication among the moving vehicles. This is where the prerequisite of edge collaboration is required between different RSUs. Our proposed CADOM facilitates the interaction between EIS-assisted RSUs and vehicles to enable seamless communication and ensure low-latency services to cater to deadline-based tasks. Moreover, it also implements a queue in case of overloading issue at a particular EIS-assisted RSU. The CADOM mechanism considers a set of vehicles $I = MEN_1, MEN_2, MEN_3, \dots, MEN_i$. Each MEN_i executes task j , where, $j \in J = 1, 2, \dots, j$. A single MEN_i generating a computation-intensive task ' j ' is characterized by the tuple: $T(MEN_i, j) = D_{MEN_i, j}^s, \partial_{MEN_i, j}, t_{MEN_i, j}^{max}$. The size of the perception task $T(MEN_i, j)$ is computed with the following parameters: $D_{MEN_i, j}^s, \partial_{MEN_i, j}$, and $t_{MEN_i, j}^{max}$, where $D_{MEN_i, j}^s$ is the data size (in bits) of the task j for vehicle user MEN_i , $\partial_{MEN_i, j}$ is the computation capability for task completion (in CPU cycles), and $t_{MEN_i, j}^{max}$ is the maximum threshold latency to complete the task. Every task $T(MEN_i, j)$ of the vehicular user i can be processed on to three tiers, namely local node processing, EIS-assisted RSUs, and cloud-assisted MBS. We consider the binary decision variables to elect the offloading and transmission mode for the task " j " of the vehicular user " i ." For the MEN_i to take an offloading decision, we take binary decision variable $d_o(MEN_i, j) = 0, 1$ where $d_o(MEN_i, j) = 1$ signifies that the task $T(MEN_i, j)$ is processed locally on the vehicle i , and $d_o(MEN_i, j) = 0$ denotes the task to be offloaded to EIS-assisted RSUs or cloud-assisted MBS. Once the decision for offloading is taken by the MEN_i for task " j ," the transmission mode is designated using another binary decision variable, i.e., $t_{modei, j} = 0, 1$. If $t_{modei, j} = 0$ is selected then, cellular mode is activated and the " j " task is transmitted to the cloud-assisted MBS, and the $t_{modei, j} = 1$ indicates that Wi-Fi mode is chosen which implies that the task is performed by EIS-assisted RSUs.

We define a geographical area as G1, having a fewer vehicle in the coverage area r_d of EIS-assisted RSU. To keep the generalization intact, we assume that there is a single vehicle for our convenience and we conduct our analysis with different scenarios of task offloading. Moreover, we assume that there are homogeneous tasks [35]. Consider that the vehicular user i having a task $T(i, j)$ and the task of the n th vehicle is $T(n, k)$, therefore, when $j = k$, we call them homogeneous tasks. Each computation task $T(MEN_i, j)$ of the vehicle i can either be computed locally or offloaded to the higher

hierarchy of computing resources (EIS-assisted RSU/ cloud-assisted MBS). For task $T(MEN_i, j)$ originating from the MEN_i , the binary decision variable $d_o(MEN_i, j) \in 0, 1$ takes the decision. If $d_o(MEN_i, j) = 1$, then we conclude that the local resources are sufficient enough for local computation, else $d_o(MEN_i, j) = 0$ indicates that the task is to be offloaded to an EIS-assisted RSU. The time taken by the task is defined as local processing delay, $p_d^l(MEN_i, j)$. Further, the mode of transmission gets selected once the on-board task has been designated to offload to either on EIS-assisted RSU or cloud-assisted MBS. We take another decision variable $t_{modei,j} \in 0, 1$ for denoting the transmission mode, where $t_{modei,j} = 1$ indicates that the task requires to be offloaded to closest EIS-assisted RSU as the task size $D_{MEN,j}^s$ is above the storage ($S_E \gg S_l$) and computing capacity $C_E \gg C_l$ of the MEN_i . The series of sub-tasks involved in the completion of one homogenous task i.e., task offloading that comprises steps of computing and feedback from the closest EIS-assisted RSU or cloud-assisted MBS to the MEN_i , is now referred to as global latency of the task. The global latency according to the CADOM is divided into following parts, namely selection time, queuing time, caching time, execution time, and feedback time which are elaborately defined in the further sub-sections. When the perception task $T(MEN_i, j)$ is initiated, the EIS-assisted RSU taking the minimum transmission time is selected, given the EIS-assisted RSU is within the geographical area G1. The minimum transmission time from a MEN_i to any SEN_n (i.e., EIS-assisted RSU) at time instance t is denoted as $Trans^{i \rightarrow n}(t)$, which is the latency time to offload the task to the closest EIS-assisted RSU, and this also refers to the selection time of EIS at time instance t denoted as $T_{slctn}^{i \rightarrow n}(t)$.

Communication Model

In our CADOM mechanism, we consider uplink communication technologies such as UDP, TCP/IP protocols, and Wi-Fi for V2X transmissions. For executing the computation-intensive tasks on the road or lane navigation, we utilize UDP for link establishment between the vehicular user and MBS. If a vehicular user MEN_i is trying to perform computation task T with the cloud-assisted servers, then different frequency bands are allocated to each link. We assume that there is no interference between the vehicles, and therefore, we calculate $\gamma_i^{t_{mode}} = [(t_{pi}^{t_{mode}} * cg_i^{t_{mode}})/\delta^2]$, where $\gamma_i^{t_{mode}}$ represents signal-to-interference plus noise ratio (SINR) from MEN_i to MBS, and $t_{pi}^{t_{mode}}$ indicates transmission power of vehicle user MEN_i . $cg_i^{t_{mode}}$ represents channel gain from MEN_i to MBS, and δ^2 indicates $\delta = awgn(x, SINR)$, i.e., $awgn$ is added to the signal x . We calculate the rate of the uplink transmission from MEN_i to the MBS on the basis of the Shanon-Hartley formula as follows: $T_r^{t_{mode}} =$

$b_i^{t_{mode}} \log_2 (1 + \gamma_i^{t_{mode}})$ where $b_i^{t_{mode}}$ denotes the cloud-assisted MBS bandwidth allocated to MEN_i . We consider that each EIS-assisted RSU has been allotted a single frequency band channel. Here, we adopt socket programming technology for establishing links between the vehicular user and EIS-assisted RSUs. Therefore, when numerous vehicular users try to perform a computation task with the same EIS-assisted RSU, there will be competition for the channel band. This will lead to intra-RSU interference, and hence, we utilize Wi-Fi as the underlying wireless communication technology where the bandwidth is high enough to cater multiple requests. In a similar way, we calculate the SINR from MEN_i to SEN_n denoted as

$$\gamma_i^{eis,n} = \frac{t_{pi}^{eis,n} * cg_i^{eis,n}}{\sum_{v \neq i, v \in MEN_i} t_{pv}^{eis,n} * cg_v^{eis,n}} \quad (2)$$

where $cg_i^{eis,n}$ denotes the channel gain from vehicular user MEN_i to SEN_n , and $t_{pi}^{eis,n}$ denotes the user transmission power. $\sum_{v \neq i, v \in MEN_i} t_{pv}^{eis,n} * cg_v^{eis,n}$ denotes the interference between other MEN_i and interference of user v to MEN_i . Note: $\gamma_i^{t_{mode}} > \gamma_i^{eis}$ which indicates $SINR_{MBS} > SINR_{EIS}$, and hence there will be no interference when MEN_i chooses MBS over EIS.

Computation Model

To handle the computation-intensive tasks, vehicular user can compute the task locally on OBUs or offload the computation task to the nearest EIS or cloud. The total processing delay can be calculated as

$$p_d = t_d + c_d + b_d \quad (3)$$

where p_d is the total processing delay, t_d is the uplink transmission delay, c_d is the computation delay, and b_d is the backhaul delay.

Local Processing Delay

Local processing refers to the computational tasks that are performed on the OBUs of vehicles itself, without relying on external computing resources. It involves executing algorithms, running ADAS applications, and performing data processing tasks directly on the vehicles. When a vehicular user MEN_i chooses vehicular server to perform the task j , the total processing delay can be calculated as follows:

$$p_d^l(MEN_i, j) = \frac{\partial_{MEN_i, j}}{\tau_{MEN_i, j}^l} \quad (4)$$

where $p_d^l(MEN_i, j)$ is the processing delay in vehicular server, $\delta_{MEN_i, j}$ is the computational requirement of the task j in CPU cycles, $\tau_{MEN_i, j}^l$ is the computation capacity of local server, and $\delta_{MEN_i, j}/\tau_{MEN_i, j}^l$ is the computation delay when performing task locally.

EIS Processing Delay

When a vehicular user MEN_i computes the task j by offloading it to EIS n , the total processing delay can be calculated as follows:

$$p_d^{eis, n}(MEN_i, j) = \frac{D_{MEN_i, j}^s}{b_w^{eis, n}_{EIS}} + \frac{\delta_{MEN_i, j}}{\tau_{MEN_i, j}^{eis, n}} \quad (5)$$

where $p_d^{eis, n}(MEN_i, j)$ is the processing delay in EIS, $\delta_{MEN_i, j}$ is the computational requirement of the task j in CPU cycles, $\tau_{MEN_i, j}^{eis, n}$ is the computation capacity of RSU, and $b_w^{eis, n}_{EIS}$ is the uplink bandwidth.

Cloud Processing Delay

When a vehicular user MEN_i chooses cloud server to compute the task j , the task data is then initially transmitted to the MBS where it gets forwarded to the cloud. We calculate the total processing delay of the vehicular user i as follows:

$$p_d^c(MEN_i, j) = \frac{D_{MEN_i, j}^s}{b_w^c_{EIS}} + (D_{MEN_i, j}^s + L_o) \cdot \rho \quad (6)$$

where $p_d^c(MEN_i, j)$ is the processing delay in cloud server, ρ is the transmission delay from MBS to cloud-assisted servers, and L_o is the data size computed from the cloud-assisted server. Note: The total processing delay relies on two factors, one is the offloading decision d_o , and the second is the transmission mode $t_{modei, j}$. Therefore, the total computational latency is as follows:

$$p_d(MEN_i, j) = d_o \cdot p_d^l(MEN_i, j) + (1 - d_o) \left[t_{modei, j} \cdot p_d^{eis, n}(MEN_i, j) + (1 - t_{modei, j}) \cdot p_d^c(MEN_i, j) \right] \quad (7)$$

The offloading decision d_o is a binary variable which indicates whether to offload an incoming task or not. The transmission mode determines the bandwidth, distance, and other characteristics of the communication channel which

are beneficial in calculating the global latency for a vehicular user MEN_i .

Metrics of Performance Analysis

This section provides a detailed formulation of the transmission time, execution time, and feedback by modeling the EIS working using the $M/M/I/N/FCFS$ queueing model in our proposed CADOM mechanism. We have used the $M/M/1/N/FCFS$ queueing model because of its ability to efficiently capture and analyze the task arrival dynamics, service processing, and performance of the system under precise conditions. It also provides analytical expressions for important performance metrics such as the average queue length and waiting time. Additionally, an optimization model is presented with the objective of minimizing the network latency. The information about PIoV includes all the static and mobile edge nodes such as EIS-assisted RSUs and vehicles, respectively, and the links for communication that connect them are kept in the DT deployed in the cloud-assisted MBS. Moreover, a Q matrix of size $k * a$ is taken to store the digital information of EIS. The D_{APIoV} of EIS can be expressed as follows:

$$Q(t) = \begin{bmatrix} q_{11}(t) & q_{12}(t) & \dots & q_{1a}(t) \\ q_{21}(t) & q_{22}(t) & \dots & q_{2a}(t) \\ q_{k1}(t) & q_{k2}(t) & \dots & q_{ka}(t) \end{bmatrix}_{k \times a} \quad (8)$$

k represents the quantity of EISs, and a denotes the number of features of individual EIS. For a convenient study, the first column represents transmission time, the second column represents computing time, and the third column represents the caching resources of the Q matrix at time instance t . Hence, we have

$$\begin{aligned} Q_1(t) &= [q_{11}(t) \ q_{21}(t) \ \dots \ q_{k1}(t)]^T \\ Q_2(t) &= [q_{12}(t) \ q_{22}(t) \ \dots \ q_{k2}(t)]^T \\ Q_3(t) &= [q_{13}(t) \ q_{23}(t) \ \dots \ q_{k3}(t)]^T \end{aligned} \quad (9)$$

Owing to the absorption and scattering phenomena, there is an inevitable loss in the wireless channel during the propagation of electromagnetic information. This loss of data during the transmission causes deviations between the DT data and the real-time data. We represent these deviations in the matrix $\Delta Q(t)$ as follows:

$$\Delta Q(t) = \begin{bmatrix} \Delta q_{11}(t) & \Delta q_{12}(t) & \dots & \Delta q_{1a}(t) \\ \Delta q_{21}(t) & \Delta q_{22}(t) & \dots & \Delta q_{2a}(t) \\ \Delta q_{k1}(t) & \Delta q_{k2}(t) & \dots & \Delta q_{ka}(t) \end{bmatrix}_{k \times a} \quad (10)$$

Correspondingly, for preserving the uniformity, the three columns of $\Delta Q(t)$ represent EIS deviations in the transmis-

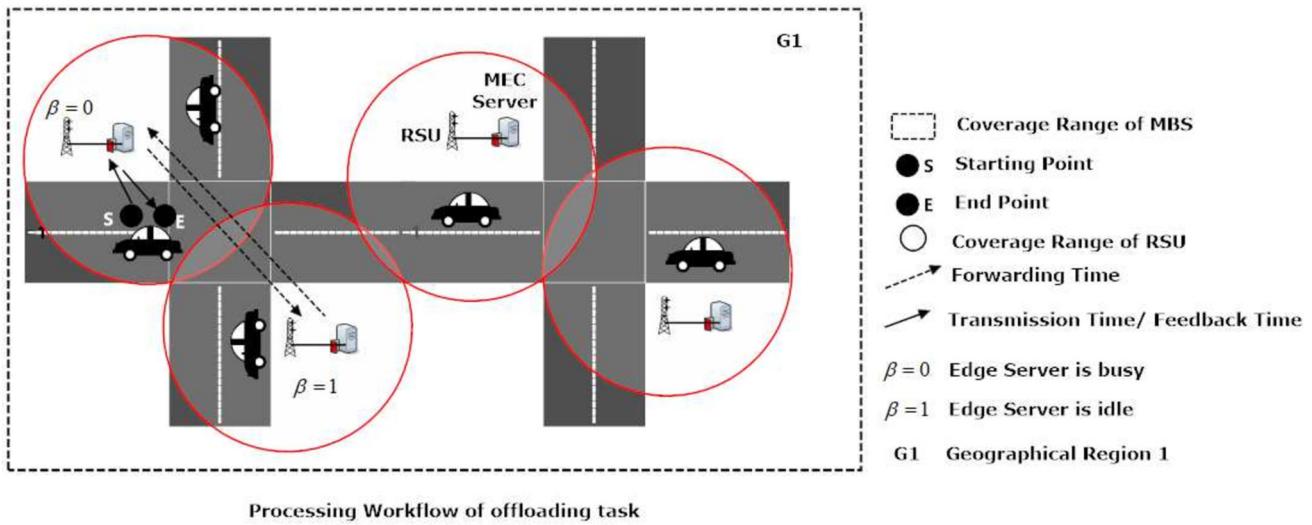


Fig. 4 Case 1-workflow of EIS within G1

sion, computing, and caching resources, stated as follows:

$$\begin{aligned}\Delta Q_1(t) &= [\Delta q_{11}(t) \ \Delta q_{21}(t) \dots \Delta q_{k1}(t)]^T \\ \Delta Q_2(t) &= [\Delta q_{12}(t) \ \Delta q_{22}(t) \dots \Delta q_{k2}(t)]^T \\ \Delta Q_3(t) &= [\Delta q_{13}(t) \ \Delta q_{23}(t) \dots \Delta q_{k3}(t)]^T\end{aligned}\quad (11)$$

Figures 4 and 5 show the DT-assisted CADOM using different transmission modes. We have taken two case scenarios to strengthen our proposed CADOM: *Case 1*: If one EIS-assisted RSU gets overloaded with a threshold number of computation requests of MENs, it forwards to another available ($\beta = 1$) EIS-assisted RSU within the same geographical area *G1*. *Case 2*: If all EIS-assisted RSUs are busy executing the computation tasks in *G1*, and the MEN has moved out of the coverage range of *G1*, then the available EIS-assisted RSUs in *G2* will connect with the MENs.

When the task size is above the computing threshold of the local processing capability, the MENs need to offload to either EIS or cloud-assisted MBS. In view of reducing the response latency time of the task completion, the MENs tend to select the best possible EIS, i.e., EIS with the utmost ideal communication environment available among several other alternative EISs.

According to CADOM, the global latency of the user request/task to offload from the vehicle and receive the result back in the vehicle can be divided into four parts:

$$\begin{aligned}G_{lat}(t) &= Real_{trans}^{MEN_i \rightarrow SEN_k}(t) + mT_{queue}^{SEN_k}(j) + T_{cache}^{SEN_j} \\ &\quad (t) + T_{exe}^{SEN_j}(j) + T_{forw}^{SEN_k \rightarrow SEN_j}(t) + T_{feed}^{SEN_i \rightarrow MEN_i}(t)\end{aligned}\quad (12)$$

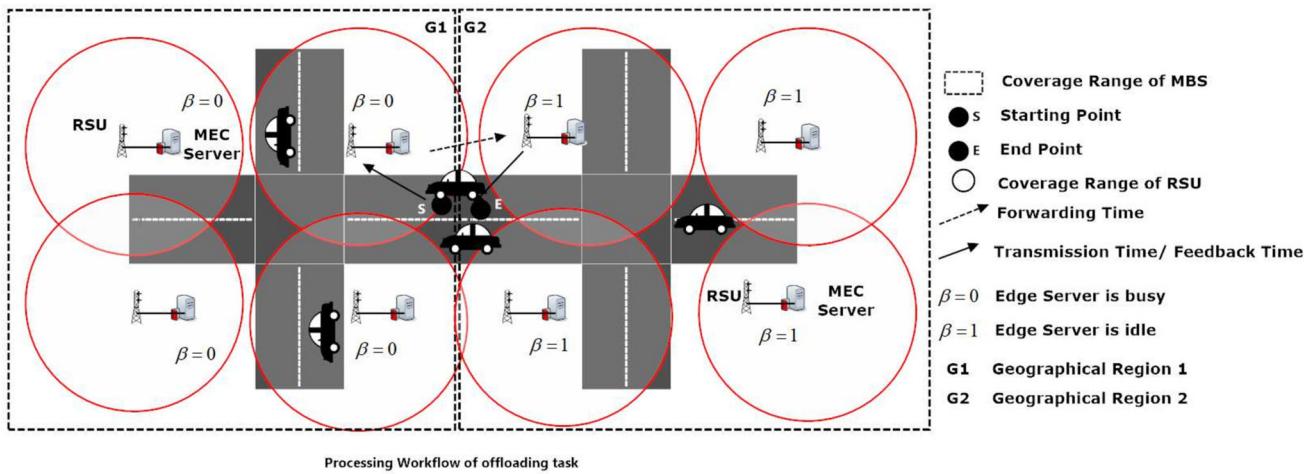


Fig. 5 Case 2-workflow of EIS across G1 and G2 when all the servers in G1 are busy

where $Real_{trans}^{MEN_i \rightarrow SEN_k}(t)$ stands for transmission time taken by MENs to offload data to static edge-node SEN_k .

$T_{forw}^{SEN_k \rightarrow SEN_j}(t)$ is the time taken to forward tasks from static edge-node k to j and SEN_k to SEN_j .

$T_{exe}^{SEN_j}(j)$ indicates the time taken to execute the task in the SEN_j .

$T_{feed}^{SEN_k \rightarrow SEN_i}(t)$ indicates the time required for SEN_k to feedback the execution result to SEN_i which is the EIS closest to the MENs as per current time.

$T_{feed}^{SEN_i \rightarrow MEN_i}(t)$ denotes the feedback time for the feedback message from $SEN_i \rightarrow MEN_i$.

Transmission Time

The transmission time is the time taken by the MEN's request to be offloaded to either EIS or cloud. The calculation of the transmission time is expressed as follows:

$$Real_{trans}^{MEN_i \rightarrow SEN_k}(t) = \frac{D_{MEN_i,j}^s}{C_{rate}^{i \rightarrow k}} \quad (13)$$

where $C_{rate}^{i \rightarrow k}$ means the rate of channel between vehicle $i \rightarrow EIS_k$ and is formulated as

$$C_{rate}^{i \rightarrow k} = q'_{k1}(t) \log_2 \left(1 + \frac{t_{p_i} * c g_{i \rightarrow k}}{N_{i \rightarrow k} + I_{i \rightarrow k}} \right) \quad (14)$$

$q'_{k1}(t)$ is the allocated bandwidth by EIS k to MEN i , t_{p_i} indicates the transmission power of vehicle user MEN_i , $c g_{i \rightarrow k}$ denotes the channel gain from vehicular user MEN_i to SEN_k , $N_{i \rightarrow k}$ indicates the noise, and $I_{i \rightarrow k}$ stands for the interference.

Selection Time

Assuming the number of EIS denoted as $SEN = SEN_1, SEN_2, \dots, SEN_n$ that are available around the vehicle within the coverage radius of r_d . Among the available EIS, we select the one with the following two constraints:

- 1) Minimum transmission time
- 2) EIS status: idle/active 0/1

The minimum transmission time is calculated in the equation as follows:

$$Real_{trans}^{MEN_i \rightarrow SEN_k}(t) = \min \left(\frac{d}{BW_{SEN_1}}, \frac{d}{BW_{SEN_2}}, \dots, \frac{d}{BW_{SEN_n}} \right) \quad (15)$$

where d is the distance of the coverage range of radius r_d and BW is the available bandwidth of RSU.

$$Real_{trans}^{MEN_i \rightarrow SEN_k}(t) = Est_{trans}^{MEN_i \rightarrow SEN_k}(t) + \Delta trans^{MEN_i \rightarrow SEN_k}(t) \quad (16)$$

where $Real_{trans}^{MEN_i \rightarrow SEN_k}(t)$ is the real-world transmission time which is calculated by $Est_{trans}^{MEN_i \rightarrow SEN_k}(t)$, and the estimated transmission time and $\Delta trans^{MEN_i \rightarrow SEN_k}(t)$ is the deviation within the DT data and actual data.

Caching Time

The request can be seamlessly addressed using DT software. Thus, it is essential to save the software that supports vehicle tasks. Nevertheless, it is not necessary for every EIS to cache their DTs in an activated state for the reasons of judicious use of storage resources. EIS i can always download the DT from other EIS j . Therefore, to indicate the state of the DT in the EIS, we employ a binary variable s_j represented as follows:

$$\begin{cases} s_j = 0 & DT_{status} \text{ is passive} \\ s_j = 1 & DT_{status} \text{ is active} \end{cases}$$

When $s_j = 1$, it requires no loading time, and if the $s_j = 0$, then it requires some loading time.

Forwarding Time

Assuming that the immediate EIS k near the vicinity of the vehicle is busy in geographical area $G1$, so the time taken to offload the request outside its coverage area r_d to another EIS j , in geographical area $G2$, is defined as the forwarding time, which is calculated as follows:

$$T_{forw}^{SEN_k \rightarrow SEN_j}(t) = d_{k \rightarrow j} / \Theta_{opt} \quad (17)$$

where $d_{k \rightarrow j}$ denotes the distance between EIS k to EIS j that are connected by an optical fiber, and Θ_{opt} is the rate of the transmission through the optical fiber (in bps). The distance between EIS can be calculated as follows:

$$d_{k \rightarrow j} = |long_k(t) - long_j(t)| + |lat_k(t) - lat_j(t)| \quad (18)$$

where $(long_k(t), lat_k(t))$ and $(long_j(t), lat_j(t))$ are the co-ordinates of EIS k and EIS j , respectively.

Execution Time

The execution of the task at the EIS depends upon the computing power at SENs. The EIS j computing ability's accurate value is stated as

$$q_j(t) = \bar{q}_j(t) + \Delta q_j(t), j \in MEN \quad (19)$$

where $q_j(t)$ is the real-time computing resource (in CPU cycles) required of EIS j , $\bar{q}_j(t)$ is the estimated computing resource at the static edge server, and $\Delta q_j(t)$ is the computing resource gap between the real-world entities and the DT. With this, we can calculate the real execution time given as follows:

$$T_{exe}^{SEN_j}(j) = \bar{T}_{exe}^{SEN_j}(j) + \Delta T_{exe}^{SEN_j}(j) \quad (20)$$

$$T_{exe}^{SEN_j}(j) = \frac{\tau_{MEN_i,j}^{eis,n}}{\bar{q}_j(t) + \Delta q_j(t)} \quad (21)$$

$\tau_{MEN_i,j}^{eis,n}$ is the computation capacity of the EIS j , $j \in SEN$.

Feedback Time

The EIS i in geographical region $G1$ sends the feedback results to the EIS j in geographical location $G2$ through the wireless channel, and the feedback result coming from the EIS j is expressed as

$$q'_{i1}(t) = q_{i1}(t) + \Delta q_{i1}(t) \quad (22)$$

The feedback time can be attained as follows:

$$\begin{aligned} T'_{feed}^{SEN_{j \rightarrow i}}(t) &= T_{feed}^{SEN_{j \rightarrow i}}(t) + \Delta T_{feed}^{SEN_{j \rightarrow i}}(t) = \\ &= \frac{q_{i1}(t) + \Delta q_{i1}(t)q_{i1}(t) - q_{i1}(t)\Delta q_{i1}(t)}{q_{i1}(t)(q_{i1}(t) + \Delta q_{i1}(t))} \times \\ &\quad \frac{d}{\log_2 \left(1 + \frac{t_{-p_i} c g_{j \rightarrow i}}{N_{j \rightarrow i} + I_{j \rightarrow i}} \right)} \end{aligned} \quad (23)$$

where $T_{feed}^{SEN_{j \rightarrow i}}(t)$ and $\Delta T_{feed}^{SEN_{j \rightarrow i}}(t)$ represent the estimated time for feedback and the difference, t_{-p_i} signifies the power of transmission, $c g_{j \rightarrow i}$ signifies the wireless channel's gain between EIS j and vehicle, and $N_{j \rightarrow i}$ specifies the noise. $I_{j \rightarrow i}$ is the interference.

Queueing Time

Since there is an assumption that each SEN_j allows a single user at a time, the computability provided by the SEN_j has hardware configuration limitations. Thereby, the process flow of SEN_j for processing the task is discussed

as $M/M/1/N/FCFS$ queueing model based on queueing theory. The upper limit is N for the computation capability of EIS j .

$$T_{queue}^{SEN_k}(j) = \bar{T}_{queue}^{SEN_k}(t) + \Delta T_{queue}^{SEN_k}(t) \quad (24)$$

where $T_{queue}^{SEN_k}(j)$ is the real-world queue time which is calculated by $\bar{T}_{queue}^{SEN_k}(t)$, and the estimated queue time and $\Delta T_{queue}^{SEN_k}(t)$ are the deviation within estimated time and the real-world queue time.

Vehicular Satisfaction Index

We also utilize the DT_{PIOV} for a cluster of MENs to determine the resource requirements and communication workload of MENs. We determine the preference based on the past performance of the MENs. If an EIS-assisted RSU fails to meet the requirements of MEN, the MEN will give less prioritization to the EIS-assisted RSU. Therefore, the MENs have less probability to offload the task to that particular EIS-assisted RSU. The satisfaction of users leads the MENs to prefer those set of EIS-assisted RSUs that have been consistent in providing satisfactory QoS. Consequently, this also motivates the EIS-assisted RSUs to improve their services to MENs, thereby maximizing their long-term gains [68]. In the DT_{PIOV} system, we have components such as SENs and MENs which have different purposes with different levels of usefulness. To maximize the benefits of the available SENs and MENs, we introduce a vehicular satisfaction index value, which indicates the service satisfaction which the MEN receives from different EIS-assisted RSUs. The vehicular satisfaction index is represented by the fraction of the aggregated vehicle satisfaction received from different SENs to the proportion of the total resources. The numerator denotes sum of difference between actual computing resources provided by EIS-assisted RSUs and the required resources for computation-intensive tasks by the vehicles considering the inclination of each vehicle is towards different EIS-assisted RSUs. The denominator is the sum of the actual computing resources provided by the RSUs. The vehicular satisfaction index (VSI) is represented as

$$VSI = \frac{\sum_{n \in SEN} \left\{ p_{i,n} * q'_{n,i}(t) - \frac{p_{i,n} * (q'_{n,i}(t))^2}{2 * \tau'_{MEN,j}} \right\}}{\sum_{n \in SEN} q'_{n,i}(t)} \quad (25)$$

where $\tau'_{MEN,j} = \frac{c * \max \tau'_{MEN,j}}{I * N}$. This formula is used to calculate the maximum amount of computing resources that can be allocated from the EIS-assisted RSUs to the MEN for a given task, where c is constant. Let $\max \tau'_{MEN,j}$ be the

atmost available and probable resource valuation provided by EIS-assisted RSUs, $\max \tau_{MEN,j}^l = \sum_{n \in SEN} \tau_{MEN,j}^l$ is formulated based on the accessible size of CPU which EIS-assisted RSU informs prior execution, and n is number of SEN. For EIS-assisted RSUs as static edge-nodes which are denoted as $SEN = SEN_1, SEN_2, \dots, SEN_n$, where $SEN_n \in SEN$. Let I be the set of vehicles that are considered mobile edge nodes MEN. $p_{i,n}$ is the preference of the vehicle i to SEN n , and $q'_{n,i}(t)$ is the real-time computing resource obtained by MEN i (*in CPU cycles*) at RSU n . The primary goal of DT_{PloV} of PE_{PloV} is to acquire additional computational resources while minimizing costs and by also considering the PE_{PloV} preferences. Consequently, the utility of the DT_{PloV} especially for PE_{PloV} is determined by U_{MEN} which represents the satisfaction of the DT_{PloV} for PE_{PloV} .

$$U_{MEN} = \sum_{n \in SEN} \left\{ W_n * r_n - \frac{(W_n * r_n)^2}{2 * \max \tau_{MEN,j}^l} \right\} - Renum\ Period \quad (26)$$

where W_n is the cumulative preference of all MENs for all SEN_n . It can be represented as $W = W_1, W_2, \dots, W_n$ that depicts the cumulative weighted preference of all SEN_n .

$$W = \frac{\sum_{n \in SEN} \partial_{MEN,j} * p_{i,n}}{\sum_{n \in SEN} \partial_{MEN,j}} \quad (27)$$

where $\partial_{MEN,j}$ denotes the computational requirement of the task in CPU cycles for computation-intensive task ' j '. Let $p_{i,n}$ be the preference of vehicle " i " over different EIS-assisted RSUs $\in SEN_n$. For EIS-assisted RSUs $\in SEN_n$, the available resources of CPU frequency are given as $\{r_n^{idle}, C_n\}$, where r_n^{idle} is the accessible computational resources and C_n calculates the rate for every unit of CPU frequency. The allocation of CPU frequency to the MENs is determined through the utilization of EIS-assisted RSUs. The CPU frequency $r = \{r_1, r_2, \dots, r_n\}$, where $r_n \leq r_n^{idle}$ is the amount of computational services EIS-assisted RSU n is ready to contribute. The term $\frac{(W_n * r_n)^2}{2 * \max \tau_{MEN,j}^l}$ is the extent of satisfaction given by the available RSU resources, and $Renum\ Period$ is utility term which considers the benefits and costs over a specific period and allows for a more informed decision-making process when allocating resources. This approach may lead to a more optimal allocation of resources in the long run. For the utility of EIS-assisted RSUs, the parameter U_n is given. The EIS-assisted RSUs aim to provide QoS to MENs at a low cost for maintaining high service satisfaction by taking more vehicular requests at a time. Therefore, the parameter U_n shows, $\forall RSU \in SEN_n$, if the EIS-assisted RSU is consistently giving services to the MENs, then high rewards are associated with the EIS-

assisted RSU. Our aim is to achieve higher satisfaction levels for MENs at a low cost. Therefore, we describe a utility function:

$$U_n = \frac{W_n * r_n}{\sum_{i \in SEN} W_i * r_i} - C_n * r_n \quad (28)$$

For the above formula, the first part can be used to determine the relative contribution of each EIS-assisted RSU to the total reward cost, and the latter part is the per cost incurred by SEN_n by providing resources. We break the problem into two parts by using the Stackelberg game. Figure 6 shows the division of the reward into two phases to gain an optimal resource management scheme (r^*, p^*) , where $r^* = (r_1^*, \dots, r_n^*)$. The first phase reward determines the amount of computing resources the EIS-assisted RSUs are willing to contribute based on the MEN requirements to maximize the EIS-assisted RSU and DT of MEN utilities. The second phase reward is to determine the resource allocation policy p^* for maximizing the VSI [69].

Problem Formulation

CADOM Minimization Function

In our proposed method, we aim to minimize the global response latency by facilitating DT-driven cooperative handshake between the SENs and thereby improving the allocation of 3C resources to the vehicle request. Moreover, the global response latency relies on whether a task can be computed locally or offloaded. If the task has to be offloaded, then which communication mode should be elected to offload the task? The portion of local and offloaded computation is denoted by $d_o(MEN_i, j)$, where if $d_o(MEN_i, j) = 1$, the whole task is computed locally, and when $d_o(MEN_i, j) = 0$, the task is to be offloaded to EIS-assisted RSU. The communication model is denoted by $t_{modei,j}$, where if $t_{modei,j} = 0$, the task is transmitted to cloud-assisted MBS, and when $t_{modei,j} = 1$, the task is transmitted to EIS-assisted RSU. The total processing delay is expressed as

$$p_d(MEN_i, j) = d_o * p_d^l(MEN_i, j) + (1 - d_o)[t_{modei,j} * p_d^{eis,n}(MEN_i, j) + (1 - t_{modei,j}) * p_d^c(MEN_i, j)] \quad (29)$$

Consequently, the optimization model for the response delay minimization is formulated as follows:

$$\begin{aligned} \text{Problem 1 } \rho 0 : & \min \sum_{d_o, t_{modei,j}} Bw_{MEN_i}^c, \tau_{MEN_i,j}^{eis,n} \\ p_d(MEN_i, j) \text{ subject to} \\ C1: & \sum \tau_{MEN_i,j}^{eis,n} \leq \tau^{eis,n} \quad MEN_i \in I \quad n \in SEN \end{aligned}$$

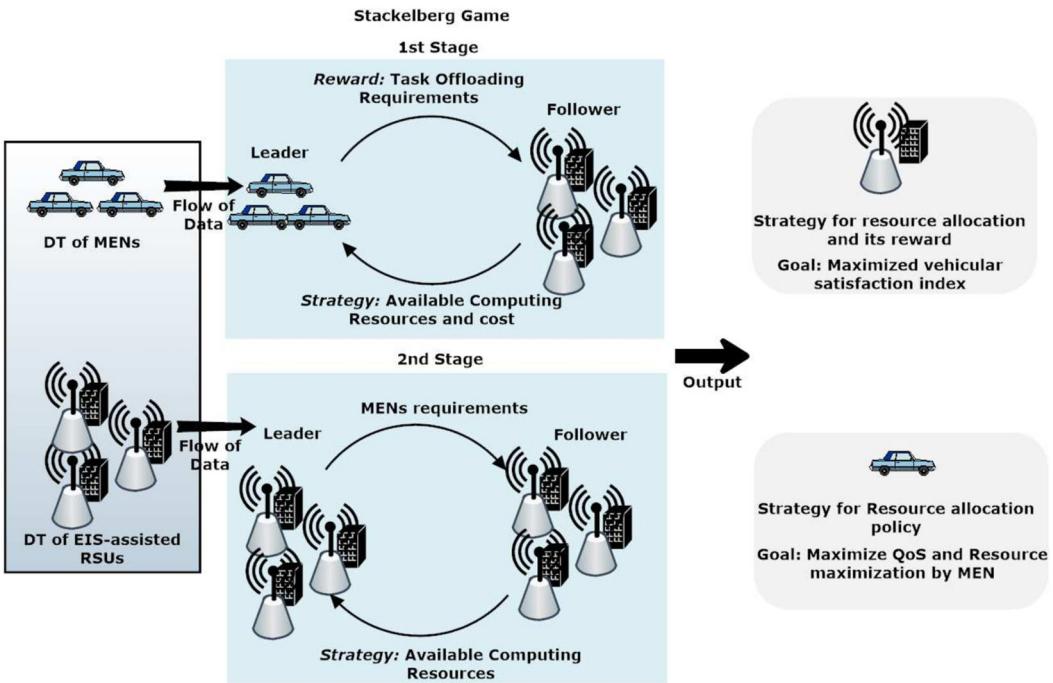


Fig. 6 Stackelberg model for task offloading in DT_{PIoV}

$$C2: \sum Bw_{MEN}^c \leq BW \quad MEN \in I$$

$$C3: \bar{q}_j(t) + \Delta q_j(t) \geq \text{argmax}(q_J(t)),$$

$$\varphi \in \varphi_1, \varphi_1, \varphi_1, \varphi_1 \dots \varphi_n$$

$$C4: p_d(MEN_i, j) \leq t_{MEN, j}^{\max} \quad MEN \in I \quad j \in J$$

$$C5: t_p^{t_{mode}} + \sum t_p^{t_{mode}} \leq t_p$$

$$C6: C_E \gg C_l$$

$$C7: S_E \gg S_l$$

$$C8: \bar{q}_j(t) + \Delta q_j(t) \geq D_{MEN_i, j}^s \quad MEN_i \in I \quad j \in J$$

$$C9: d \leq r_d$$

$$C10: Real_{trans}^{MEN_i \rightarrow SEN_k} \leq \min(\sum_{i=1}^n SEN_n)$$

Assumption There are a few assumptions that we have made in our scenarios for the simulations as follows:

$$A1: t_{modei, j} \in \{0, 1\} \quad i \in I, j \in J$$

$$A2: d_o \in \{0, 1\} \quad i \in I, j \in J$$

A3 : $\beta_{i,j} \in \{0, 1\} \quad MEN_i \in I \quad j \in J$ where $\rho_0, d_o(MEN_i, j)$ denotes offloading decision binary decision variable, while $t_{modei, j}$ denotes the decision variable of transmission mode, respectively. Furthermore, Bw_{MEN}^c denotes the MBS's bandwidth allocation, while $\tau_{MEN_i, j}^{eis, n}$ denotes the computation resource allocation. **Constraint C1** depicts the total computation resource capacity of all vehicles that do not exceed the computing and storage capacity of the EIS.

Constraint C2 denotes that bandwidth allocated to the vehicle does not exceed the total possible bandwidth at the cloud-assisted MBS. **Constraint C3** is defined to ensure that the computing ability of EIS in CPU cycles should be sufficient to handle the task from the vehicular user. **Constraint**

C4 denotes that the total propagation delay p_d of a set of tasks from the vehicular user should not exceed the maximum tolerable latency value randomly selected between (0.2 – 2 s). **Constraint C5** defines that the total power consumed for computation should not exceed the main power expenditure of collaborative EIS. **Constraint C6** and **Constraint C7** represent the computing and storage constraints of EIS that should be greater than the available local processing. **Constraint C8** is to ensure the cache space in EIS that has sufficient caching to download DT essential for executing the task. **Constraint C9** ensures that the distance of the vehicle is within a range of EIS-assisted RSU. **Constraint C10** denotes that the transmission time should be less than equal to the minimum of transmission time of all EIS within range r_d . **Assumptions A1** and **A2** denote the value of transmission mode and the coefficient for deciding the offload strategy. **Assumption A3** is a decision strategy that defines if SEN has the computation ability for the given task. We divide the optimal problem ρ_0 into two suboptimal problems to solve and find a solution for ρ_0 . **Sub-optimal model 1:** ρ_2 is assigned to minimize the section time of the EIS, i.e., optimally allocating the communication resource.

$\rho_2 : \min T_{select}^{MEN_i \rightarrow SEN_k}(t)$ where constraints are as follows:

$$C1: d \leq r_d$$

$$C2: \bar{q}_j(t) + \Delta q_j(t) \geq \text{argmax}(q_J(t))$$

$$C3: \sum Bw_{MEN_i}^c \leq B_W$$

$$C4: \sum \tau_{MEN_i, j}^{eis, n} \leq \tau^{eis, n} \quad MEN_i \in I \quad n \in SEN$$

$$C5: t \leq \min(\sum_{i=1}^n SEN_n)$$

Sub-optimal model 2: ρ_3 is built to cooperatively enhance caching and computation resources.

$$\rho_3 : T_{queue}^{SEN_k}(j) + T_{cache}^{SEN_j}(t) + \\ T_{exe}^{SEN_j}(j) + T_{forw}^{SEN_k \rightarrow SEN_j}(t) + T_{Feed}^{SEN_i \rightarrow MEN}(t)$$

where constraints are as follows:

- C1 : $\sum \tau_{MEN_i,j}^{eis,n} \leq \tau^{eis,n}$, $MEN_i \in I$, $n \in SEN$
- C2 : $p_d(MEN_i, j) \leq t_{MEN_i,j}^{\max}$, $MEN_i \in I$, $j \in J$
- C3 : $\bar{q}_j(t) + \Delta q_j(t) \geq D_{MEN,j}$, $MEN_i \in I$, $j \in J$
- C4 : $d \leq r_d$
- C5 : $S_E \gg S_l$

Utility Maximization Function Based on Reward Mechanism

Additionally, we also maximize the individual utilities of the PIoV in the DT_{PIoV} as discussed in *Section V*. Further, to achieve the goal of optimal RA between the EIS-assisted RSU and MENs, we initially determine the contribution of CPU resource allocation given by every EIS-assisted RSU based on the needs and preferences of MENs. We formulate and analyze the optimal solution based on a Stackelberg game where there are two players: a leader and a follower. The leader is the first player to decide, and the follower responds to the leader's decision. In our scenario, DT of MEN acts as a leader and EIS-assisted RSU as a follower. In order to maximize the utility and boost the EIS-assisted RSU for contributing better computing resources, the leader announces a reward R for EIS-assisted RSU by telling the requirement for task offloading and vehicle preference information after the EIS-assisted RSU will specify their respective strategies. The strategy includes determining the amount of computing resource, i.e., r_n , the EIS-assisted RSUs are willing to contribute based on the rewards of MENs. The game can be formulated as follows:

$$P1 : \max_R U_{MEN}(R) : Leader$$

$$\begin{aligned} & \max_{r_n} U_n(r_n, r_{-n}) : Follower \\ & \text{s.t. } r_n \leq r_n^{idle} \forall n \in SEN \end{aligned}$$

where r_n^{idle} states the at-most computing resource that the EIS-assisted RSU n will be able to contribute. Let the RSU's strategies profile be represented by $r = r_1, \dots, r_{SEN_n}$. Then, $r_{-n} = r_1, \dots, r_{n-1}, r_{n+1}, \dots, r_N$ represents strategy of EIS-assisted RSUs exclusive of r_n , the game theory consists of 2 stages:

1st stage: DT of vehicles in the first part announces rewards for the EIS-assisted RSUs, and in the subsequent part, all EIS-assisted RSUs ensure the computational resources quantity that they can provide in order to maximize their respective utility. *2nd stage:* Further, in the second part, they establish a noncooperative game, i.e., there is

a Nash equilibrium (*NE*) as the stability of the complete game is attained, only if the profile of the strategy consisting of EIS-assisted RSU's strategy is stable. *Description 1 (NE):* A set of strategies for EIS-assisted RSU is denoted as $r^* = (r_1^*, \dots, r_n^*, \dots, r_N^*)$ is said to be NE if for any EIS-assisted RSU n

$$U_n(r_n^*, r_{-n}^*) \geq U_n(r_n, r_{-n}^*) \forall r_n > 0. \quad (30)$$

The above equation proves that EIS-assisted RSU cannot leverage any extra profits by altering its existing strategy. For *NE*, we have formulated the best response strategy for every EIS-assisted RSU, because EIS-assisted RSU n can use strategy r_n^* only if other EIS-assisted RSUs use their respective optimal strategy. The *NE* is reached when every RSU selects a utility maximization strategy, considering the fixed strategies of other RSUs. In this state, no RSU can improve its outcome by changing its RA, assuming other RSUs keep their strategies static.

Algorithms

According to the task details and the number of available resources, we first propose a selection time algorithm to allocate communication resources optimally and then determine the best response strategy for the RSU.

Algorithm 1 EIS k selection time algorithm on the basis of triads.

```

Data:  $Q_1(t), \Delta Q_1(t), k$ , where  $r_d$  is coverage radius of EIS;  $k$  is number of EIS-assisted RSUs
Result: EIS  $k$ 
1 initialization
  for  $i \leftarrow 1$  to  $k$  do
     $Real_{trans}^{MEN_i \rightarrow SEN_k}(t) = Est_{trans}^{MEN_i \rightarrow SEN_k} +$ 
     $\Delta trans^{MEN_i \rightarrow SEN_k}(t)$ ;
    if  $Real_{trans}^{MEN_i \rightarrow SEN_k} \leq MaxLat$  &&  $\beta = 1$  then
      save EIS  $k$  into  $\epsilon$ ;
      break;
    end
  end
5 end
6  $MinTrans = Real_{trans}$ 
  if  $\epsilon \neq null$  then
    for  $i \leftarrow 1$  to  $k$  do
      if  $Real_{trans}^{MEN_i \rightarrow SEN_k} \leq MinTrans$  &&  $\beta = 1$  then
         $MinTrans = Real_{trans}^{MEN_i \rightarrow SEN_k}$ ;
        break;
      end
    end
11 end
12 end
13 return EIS  $k$ 

```

Selection of Best Available EIS-Assisted RSU Based on Triads, i.e., Transmission Time, Computing, and Caching

The selection task is primarily resolved by following a series of steps, including task offloading, task execution, and response to the results. These steps break down the original model (optimization) into two different sub-optimal models. *Algorithm 1* is specifically created to address *sub-optimal model 1* and enhance the efficiency of task offloading by minimizing transmission delays. We initialize an empty set which will contain the available EIS k . The $Real_{trans}^{MEN_i \rightarrow SEN_k}(t)$ calculates the real transmission time between a vehicle i and EIS k at time instance t . This also considers the deviation $\Delta_{trans}^{MEN_i \rightarrow SEN_k}(t)$ within the DT data and actual data. To select an appropriate EIS k , two conditions are checked. First, the real transmission time should be less than the maximum tolerable latency, and secondly, the EIS k should be available, i.e., $\beta = 1$. If the two conditions are satisfied, the respective EIS k is initialized into the empty set \in . This selection is set as a precedence for any upcoming selections of EIS and is, therefore, designated as a set containing EIS having *MinTrans*.

If list already contains some set of EIS, then again, two conditions are checked. First, the real transmission time $Real_{trans}^{MEN_i \rightarrow SEN_k}(t)$ between a vehicle i and EIS k should be less than *MinTrans*, and EIS k should be available, i.e., $\beta = 1$. If the two conditions are satisfied, variable *MinTrans* will hold an updated EIS k suitable for communication. On the basis of Algorithm 1, we can recognize and allocate the communication resource, i.e., the EIS-assisted RSU that possesses optimal communication conditions and abundant resources to facilitate task offloading. This allocation process effectively minimizes transmission delays, leading to improved efficiency. Upon receiving the assigned task by EIS k present in geographical area $G1$, and by the time the vehicle moves out of $G1$ to $G2$, the request is forwarded to other EIS j in $G2$. The DT facilitates this collaboration of EIS $k \rightarrow j$ that updates the present state of other EISs, i.e., their available resources and whether the DT for executing the task has been cached or not.

Once the tasks are queued, EIS-assisted RSU j processes them. The feedback is then transmitted back to the vehicle through a feedback link. Throughout this process, the vehicle's mobility is taken into account, where it keeps moving during the offloading process. Further, Algorithm 2 follows the forwarding principle when the MEN is crossing between two different geographical regions as shown in Fig. 5, where the EIS k of $G1$ forwards the request to EIS j of $G2$ to satisfy the connection request. Firstly, the $M[]$ stores the non-overlapping EIS of $G1$ and $G2$ and then again applies the selection algorithm on $M[]$. Algorithm 2 returns the suitable EIS after implementing the selection algorithm.

Algorithm 2 Forwarding algorithm of user request.

```

Data: nEISG1[ ], nEISG2[ ]
14 M[ ] = nEISG2[ ] - nEISG1[ ] //List of EIS in G2 without
   overlapping
   EIS = Selection_time(M[ ])
   return EIS

```

Second Phase Nash Equilibrium for Reward-Based Mechanism

Definition 2 (Best Response Strategy): For each EIS-assisted RSU, best response strategy $Sr_n(r_{-n})$ can be calculated when it maximizes its utility $U_n(r_n, r_{-n}) \forall r_n \geq 0$ and deriving it w.r.t r_n to get the *NE* for the second stage. We derive the U_n wrt r_n :

$$\begin{aligned} \frac{\partial U_n}{\partial r_n} &= -\frac{W_n \sum_{i \in SEN} W_i r_i - W_n^2 r_n}{(\sum_{i \in SEN} W_i r_i)^2} R - C_n \\ &= -\frac{W_n^2 r_n}{(\sum_{i \in SEN} W_i r_i)^2} R + \frac{W_n}{\sum_{i \in SEN} W_i r_i} R - C_n \quad (31) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 U_n}{\partial r_n^2} &= -\frac{2 W_n^2}{(\sum_{i \in SEN} W_i r_i)^2} R + \frac{2 W_n^3 * r_n}{(\sum_{i \in SEN} W_i r_i)^3} R \\ &= -\frac{2 W_n^2 \sum_{i \neq n} W_i r_i}{(\sum_{i \in SEN} W_i r_i)^3} R < 0 \quad (32) \end{aligned}$$

As the second-order derivative function < 0 , the RSU utility is strictly concave in r_n . Therefore, it comprehends that given the conditions $R > 0$ and any profile of the strategy r_{-n} , $\beta_s(r_{-n})$ is unique. Equalizing the 1st derivative to 0,

$$-\frac{W_n^2 r_n}{(\sum_{i \in M} W_i r_i)^2} R + \frac{W_n}{\sum_{i \in SEN} W_i r_i} R - C_n = 0 \quad (33)$$

$$r_n = \frac{1}{W_n} \left(\sqrt{\frac{W_n R \sum_{i \neq n} W_i r_i}{C_n}} - \sum_{i \neq n} W_i r_i \right) \quad (34)$$

If the above equation is negative, it means that its computing resources of EIS-assisted RSU n will not be accessible for the incoming task, i.e., $r_n = 0$. Else, the equation r_n is the idle response strategy of EIS-assisted RSU n . Further, if the right-hand side is greater as compared to r_n^{idle} then, the EIS-assisted RSU n provides it with the optimum response by equalizing $r_n = r_n^{idle}$. Hence, we represent the best response strategy as

$$Sr_n(r_{-n}) = \begin{cases} 0, & \frac{C_n}{W_n} \geq \frac{R}{\sum_{i \neq n} W_i r_i}, \\ \sqrt{\frac{W_n R \sum_{i \neq n} W_i r_i}{C_n} - \frac{\sum_{i \neq n} W_i r_i}{r_n}}, & 0 \leq r_n \leq r_n^{idle} \\ r_n^{idle}, & r_n \geq r_n^{idle} \end{cases} \quad (35)$$

Since the evaluation of the best response strategy is an approximation, we calculate an exclusive NE for the second part of the game. *Statement 1:* Given $R > 0$, let $r^* = (r_1^*, \dots, r_n^*)$ be the nash equilibrium value, and let $\vartheta = \{SEN_n \in SEN \mid r_n > 0\}$ represent RA for a group of EIS-assisted RSUs. We can say

- 1) $|\vartheta| > 2$
- 2) Best response strategy is modeled in a closed-form solution represented as follows:

$$r_n^* = \begin{cases} 0, & n \notin \vartheta \\ \frac{\Delta}{W_i} \left(1 - \frac{C_i * \Delta}{W_i}\right) R, & n \notin \vartheta \text{ and } r_n \in (0, r_n^{idle}) \\ r_n^{idle}, & \text{otherwise} \end{cases} \quad (36)$$

where $\Delta = \frac{|\vartheta|-1}{\sum_{j \in \vartheta} \frac{C_j}{W_j}}$

3) If $\frac{C_1}{W_1} \leq \max_{n \in \vartheta} (\frac{C_n}{W_n})$, then

4) Assume EIS-assisted RSUs are ordered as $\frac{C_1}{W_1} \leq \frac{C_2}{W_2} \leq \dots \leq \frac{C_n}{W_n}$.

Let the highest integer number be B in $[2, SEN_N]$; we

have $\frac{C_B}{W_B} < (\frac{\sum_{i=1}^B C_i}{B-1})$. Then, $\vartheta = \{1, 2, \dots, B\}$

Proof The thorough proof of this *statement 1* is comparable to [69]. The NE in this stage occurs when no vehicle or RSU can enhance its own utility by improving its allocation policy, considering the policies of others. This equilibrium reflects a state where RA is balanced across the IoV network, minimizing computational delay and energy consumption.

Stackelberg Equilibrium in the Nascent Stage

The unique NE is formulated in the second phase of the game. Given this, we want to formulate a unique Stackelberg equilibrium for the entire game. For this, the required and sufficient constraint is that the nascent stage has a unique optimal solution. *Statement 2* \exists unique Stackelberg equilibrium (R^*, r^*) where R^* maximizes DT vehicle utility over a range $[0, \infty]$, r^* where given by eq. 35. Algorithm 3 calculates the Nash equilibrium where the following values are taken as input: number of EIS-assisted RSU, cost function, and cumulative weighted preference of all SEN_n . It returns the Nash equilibrium r_n^* . In the algorithm, at first, the EIS-assisted RSUs are sorted in ascending order based on their contribution which is divided by their weight. Iterate through and find a list of all available RSU and store as ϑ , ϑ be list of EIS-assisted RSU which have computation power to do the task offloading function. We check the RSU contribution

divided by their weight is less than or equal to the weighted average of the contributions of RSUs and store the resource information in cycles per second. Finally, the optimal RA for each RSU is calculated based on whether it belongs to ϑ or not and whether its remaining period is in the range of 0 to the idle period. The resulting optimal allocation is returned as a list of values for each EIS-assisted RSU, which calculates the optimal resource allocation for a group of EIS-assisted RSUs based on their computational contributions and weights.

Algorithm 3 Nash equilibrium.

```

Data: Nash Solution ( $SEN, C, W$  );
Result: NE  $r^*$ 
15 Sort the RSU according to  $\frac{C_1}{W_1} \leq \frac{C_2}{W_2} \leq \dots \leq \frac{C_n}{W_n}$ 
// list of all available RSU and store as  $\vartheta$ 
 $\vartheta \leftarrow 1, 2$ 
for  $i \in SEN$  do
16    $i \leftarrow 3$ ;
   if  $i \leq SEN_n$  &&  $\frac{C_i}{W_i} \leq \frac{\frac{C_1}{W_1} + \dots + \frac{C_{i-1}}{W_{i-1}}}{W_i}$  then
17      $\vartheta \leftarrow \vartheta \cup \{i\}$ ;
      $i \leftarrow i + 1$ 
18   end
19 end
20 //Traverse through
21 for  $i \in SEN$  do
22   if  $i \in \vartheta$  &&  $r_{SEN_n} \in (0, r_i^{idle})$  then
23      $\Delta = \frac{|\vartheta|-1}{\sum_{j \in \vartheta} \frac{C_j}{W_j}}$ 
      $r_n^* = \frac{\Delta}{W_i} \left(1 - \frac{C_i * \Delta}{W_i}\right) R$ ;
24   end
25   if  $i \notin \vartheta$  then
26      $r_n^* = 0$ ;
27   end
28   else
29      $r_n^* = r_i^{idle}$ 
30   end
31 end
32 return  $r^* = (r_1^*, \dots, r_n^*, \dots, r_N^*)$ 

```

Performance Analysis

Simulation Setup

Simulation tests are conducted to demonstrate the practicality and relevance of DT_{PIoV} in the conjoint proposed objective of CADOM and reward-based RA mechanism. These simulations substantiate the proposed methods by using a computer of a system configuration *Intel®Xeon® CPU @ 2.20GHz with Core i7 9th Gen* and a *Nvidia GeForce GTX 1080 Ti GPU* an *Intel Core i7-10210U 2.11-GHz CPU, 8-GB RAM*, and *Windows 10* operating system. *MATLABR2023a, SUMO, OSM* are used as the software for running the simu-

lations. The key parameter values utilized in the simulation are provided in Table 3. We assume the total number of RSUs taken is 5. The coverage range of the sensor fixed on RSU is 200 m. The total number of available computing resource, i.e., $\tau_{MEN,j}^l$ is [30–60] GHz. We assume a total number of 10–15 vehicles in constant mobility at an average speed of 20–120 Km/h. The vehicles capture the real-time data by the cameras mounted on them whose sensing range is 80 m. The number of EIS-assisted RSUs that can be connected to the vehicle is 3, and the total spectral bandwidth is in the range of 10–50 MHz. The separation distance between 2 RSU is approximately 150 m. The vehicles taking input are of varying sizes between 0.3 and 12 MB at three different resolutions 720*480, 720*720, 1920*1080. The number of vehicles served by one EIS-assisted RSU (queue) is 5. The experimental simulation process is elaborated in the next subsection. Further, the DTs of vehicles and EIS-assisted RSUs depict the time-varying supply–demand of the resources. The preferences of different vehicles offload their computation-intensive tasks to RSUs.

Evaluation and Simulation Results

A series of experiments are conducted to evaluate the effectiveness of the TSTP-LSH solution for traffic flow and velocity prediction on a real traffic dataset gathered in India. The experiments were carried out on a curated dataset every 15 min for a number of intersections on the road in a day, including the volume and speed of passing traffic. One of the key parameters is accuracy, which assesses the traffic pre-

diction performance. The accuracy should be high enough to enable the traffic governors to gain comprehensive insights into future traffic scenarios. *Test1*: The first test measured the accuracy of the contemporary methods in predicting traffic flow with respect to different time intervals. The prediction showed that accuracy of traffic flow prediction using the TSTP-LSH solution was higher than that of Naive K-NN, Enhanced K-NN, and GBM at various time windows. This was accomplished because of the time-critical LSH technique's high degree of similarity-maintenance, which was applied in TSTP-LSH. After time-aware LSH search, the “most similar” days could be identified using data on average traffic flow, leading to high prediction accuracy. The results can be seen in Figs. 7, 8, and 9, where TSTP-LSH can be used for predicting the traffic flow at any point of time, i.e., peak hours or non-peak hours. This resultant accuracy makes the proposed TSTP-LSH algorithm suitable for the applicability in the traffic prediction. *Test2*: In this test, we assess the speed of the traffic for the existing methods Naive K-NN, Enhanced K-NN, and GBM with the proposed TSTP-LSH at different intervals of time including peak and non-peak hours.

According to the time slice division principle, we consider time windows in a 24-h day. The obtained experimental results are depicted in Figs. 10, 11, and 12. We witness that the accuracy of the TSTP-LSH solution's traffic speed prediction at various traffic hours outperformed that of the methods such as Naive K-NN, Enhanced K-NN, and GBM. The reason for this was that the LSH technique used in TSTP-LSH has a good similarity-maintaining quality that made it pos-

Table 3 Simulation parametric values

Parameter	Value
Number of EIS-assisted RSU	5
Number of vehicles MEN	10–15
Task size of vehicles $T(MEN_i, j)$	[0.6–60] MB
Number of available computing resource $\tau_{MEN,j}^l$	[30–60] GHz
Average vehicular speed	[20–120] Km/hr
Coverage radius of sensor fixed on EIS-assisted RSU r_d	200 m
Sensing range of cameras mounted on vehicles	80 m
Total spectral bandwidth b_w	[10–50] MHz
Distance between two RSU	150 m
Size of input data at three different resolutions $D_{MEN,i,j}^s$	720*480, 720*720, 1920*1080
Maximum tolerable latency $t_{MEN,i,j}^{max}$	[0.2–2] sec
Number of vehicles served by one EIS-assisted RSU (queue)	5
Vehicle inclination value Q_{nm}	[0,10]
Cost per CPU frequency unit C_n	[1,2]
CPU structure of EIS-assisted RSU (σ)	$\sigma = 1$
Preference of MEN for EIS-assisted RSU $p_{i,n}$	[0, 10]

Fig. 7 MAPE: predicted accuracy of traffic density vs time slices in a day

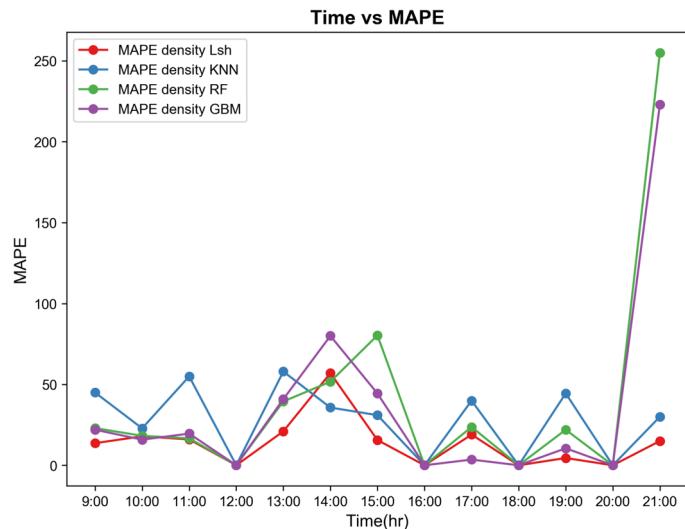


Fig. 8 MAE: predicted accuracy of traffic density vs time slices in a day

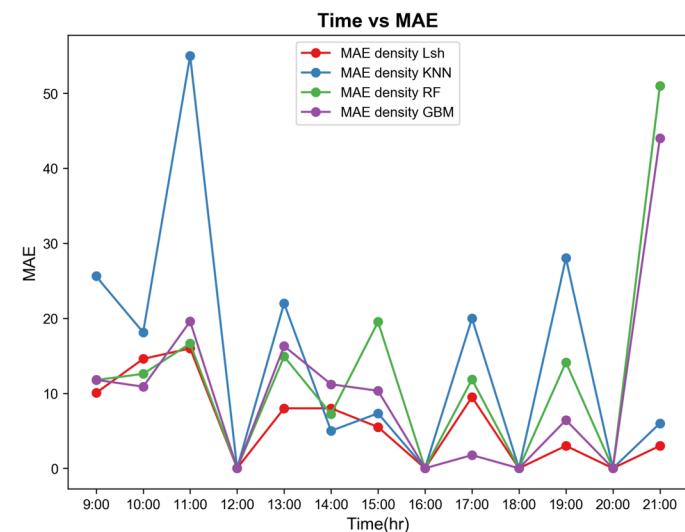


Fig. 9 RMSE: predicted accuracy of traffic density vs time slices in a day

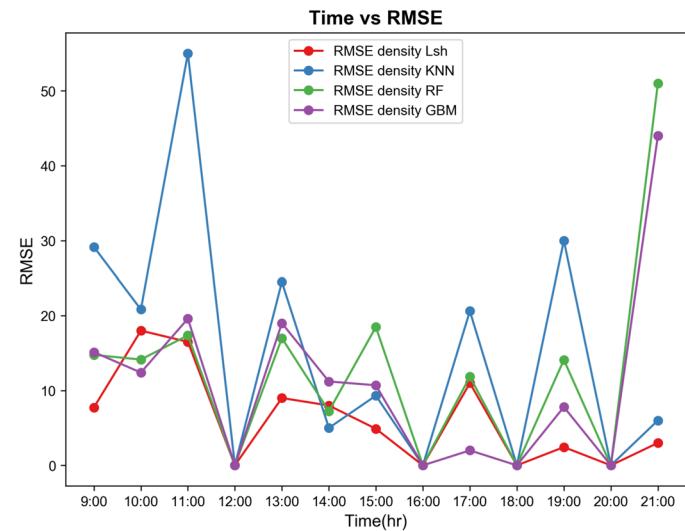


Fig. 10 MAE: predicted accuracy of traffic speed vs time slices in a day

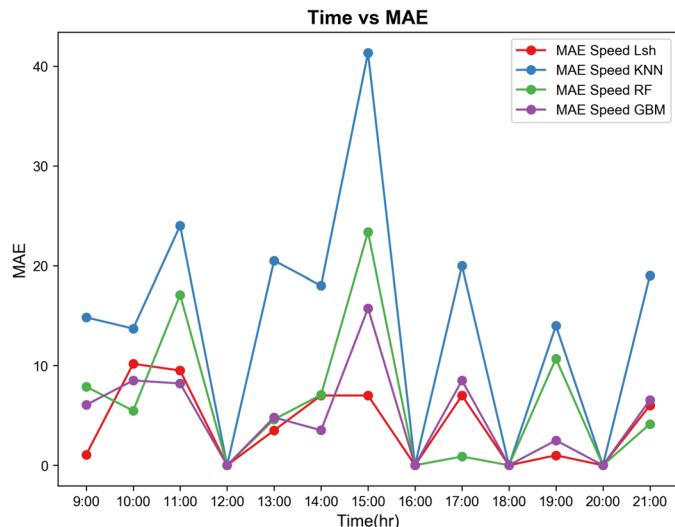


Fig. 11 MAPE: predicted accuracy of traffic speed vs time slices in a day

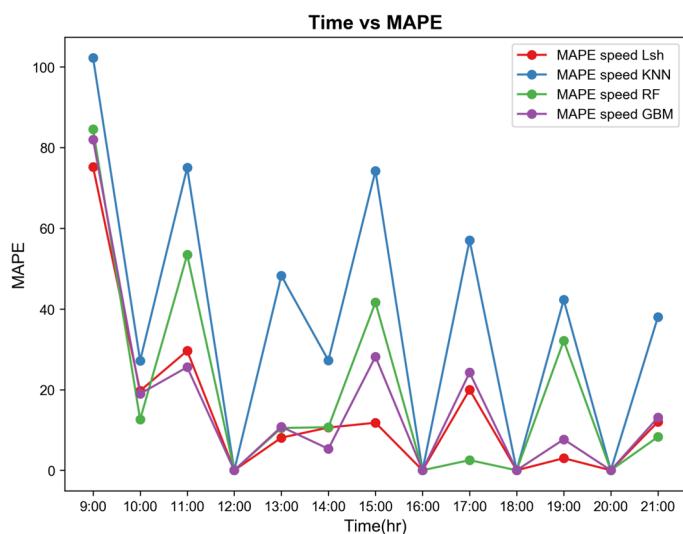
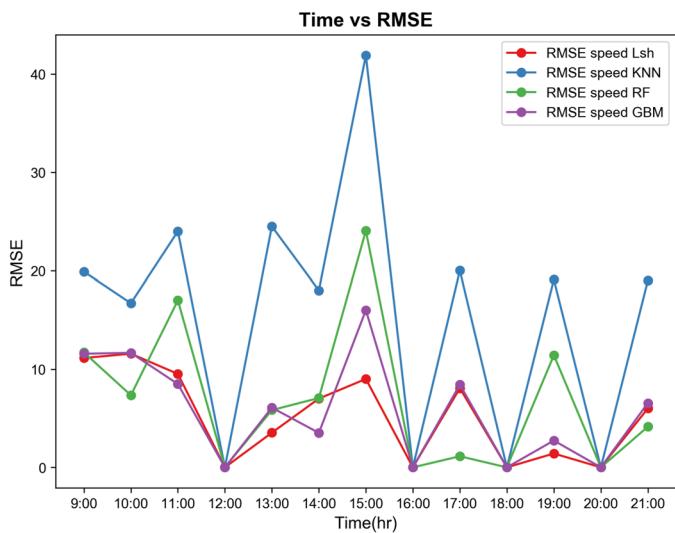


Fig. 12 RMSE: Predicted accuracy of traffic speed vs time slices in a day



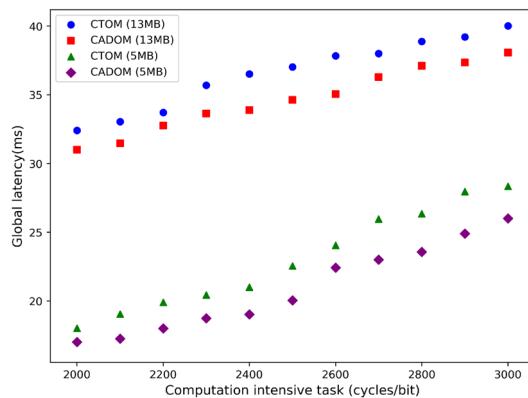


Fig. 13 The response latency of the system vs computation intensity of the task for different task offloading strategies

sible to find the “most similar traffic situations” following a time-aware LSH-technique, leading to generally excellent traffic speed forecast accuracy.

As the number of in-vehicle applications is mushrooming, there is a variation in the pixel requirements for video-related tasks such as image acquisition and object detection. Hence, this study takes into account three types of pixel resolutions: *standard resolution* ($720*480$), *high definition* ($720*720$), and *ultra-high definition* ($1920*1080$). It is important to mention that the input data size, denoted as $D_{MEN,j}^s$, differs for each of these resolutions. In this section, we provide a summary of the steps taken to evaluate the proposed scheme. For the implementation of DT, we constructed a MATLAB-based simulation model and recorded the actual outcomes of information exchange between MENs (vehicle clients) and SENs (EIS-assisted RSUs) in various scenarios/environments. We gathered data from this transaction and designated the actual latency time with DT. For without DT, we considered factors such as Shannon capacity and signal-to-interference plus noise ratio (SINR) as real-time interferences in the network dynamics while calculating transmission time and feedback time. Based on the data presented in Table 4, it can be observed that the DT assistance improves the feedback

time significantly compared to the scenarios without DT. For instance, the input video task size in (1) in Table IV, the feedback time is 0.000556193 s without DT. However, the response time sharply drops to 0.00099969 when DT assistance is there. The primary factor behind this improvement is the effective guidance and management of task scheduling enabled by the emergence of DT. This, in turn, contributes in reducing the network delay.

To demonstrate the effectiveness of our proposed CADOM, this article presents a comparison with the conventional task offloading method (CTOM). Figure 13 illustrates the comparison between the offloading schemes based on the global latency for two different input data sizes of 5 Mb and 13 Mb. From Fig. 13, we can observe that when the video input is 13 Mb, the global latency of the whole system rises in comparison to the case with a 5 MB input. Moreover, when the video input remains constant at 5 Mb, but the computation intensity increases, the global latency of both the CADOM and CTOM methods also increases. Nevertheless, the proposed CADOM method demonstrates significant improvement over CTOM in reducing the global latency of the whole system, irrespective of any fluctuations in the input data size or intensity of the computations.

Further, this article also considers certain other parameters that influence the system latency for, e.g., available wireless bandwidth bw . Figure 14 illustrates that any fluctuations in the bandwidth primarily impact the delay in transmission time. In case of the CTOM method, the MENs only upload the unprocessed input data to SENs, making them highly susceptible to b_w fluctuations. Consequently, with the constant rise in available resources of b_w , the global latency of the CTOM approach decreases significantly. For instance, we have taken a total $b_w = 65$ MHz; the overall delay of the CTOM method approaches nearer to that of the CADOM method. Nonetheless, for the proposed CADOM, if the available SENs are less in number, the queueing approach can be availed.

When we consider total $B_W = 30$ MHz as an instance, it can be observed that the delay in global latency of the

Table 4 Transmission and feedback time comparison in two scenarios (with DT and without DT) during the offloading process between MEN and SEN

S.No	Size (kB) (Client sending files)	Transmission time DT(s)	Transmission time w/o DT(s)	Feedback time(s) with DT	Feedback time(s) w/o DT
1	324	0.03	0.01	0.001	0.003
2	555	0.01	0.01	0.002	0.002
3	1535	0.01	0.02	0.001	0.002
4	4925	0.02	0.04	0	0.002
5	10,423	0.04	0.07	0	0.001
6	12,961	0.02	0.09	0	0.001
7	7562	0.02	0.04	0	0.001

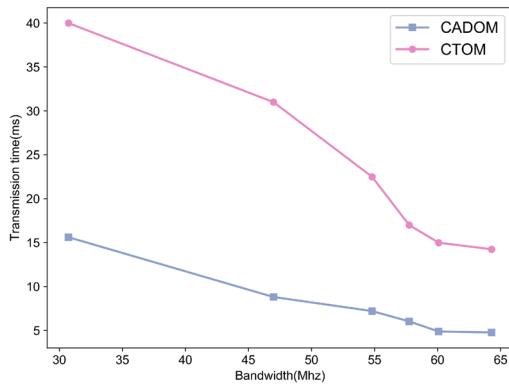


Fig. 14 The transmission time delay of the system vs bandwidth of the task for different task offloading strategies

proposed CADOM is significantly lower compared to that of the CTOM. This suggests that the performance enhancement achieved by the proposed CADOM is more pronounced than that of the CTOM, especially when the wireless B_W becomes the limiting factor in the system. Hence, we can say that our experimental results validate the proposed CADOM method to be more effective over the CTOM under the fluctuating B_W constraint. Further, we also witness that the input video size also affects the performance of the whole system leading to larger delay in global latency. In Fig 15, the graph displays the global latency of the system for various input data sizes. Across seven different input data sizes, the CADOM with DT algorithm exhibits the lowest response delay in comparison to the existing methods. Moreover, as the input size of the data increases, there is a proportional surge in the global latency. This is because when the video input is of more size, it takes more time to load and execute. This affects the feedback time in which the response is given back to the vehicle. The incorporation of DT in our scenario boosts the process by facilitating the simulation and informing the respective MENs giving real-time response with low latency.

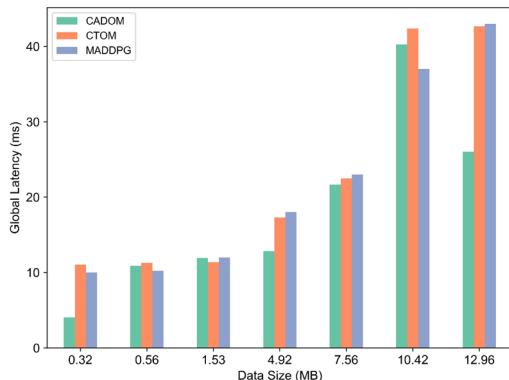


Fig. 15 The global latency of the system vs. input size of the task for with and without DT scenarios

Figure 16 illustrates the global latency of the system with varying numbers of SENs, i.e., the edge-intelligent servers EIS, to show the cooperative handshake between them. As the number of EIS increases, the system's global latency decreases continuously. This is attributed to the fact that with a fixed number of MENs and an increasing number of EIS under time instance t (i.e., same time slot), the share of computation tasks offloaded to EIS increases. Consequently, the proportion of local computing tasks performed at MENs decreases. As a result, the parallel computation time delay decreases, leading to a reduction in the latency delay. It can be observed from Fig. 16 that the CADOM method employed in this study outperforms CTOM in terms of system performance.

We further compare the influence of the techniques without DT and with DT in CADOM on transmission delay. We can conclude from Fig. 17 that our CADOM with DT technique shows less transmission delay which is indeed better in latency performance. According to Fig. 17, we analyze the change in behavior of the transmission time with varying data size. We can observe that CADOM indeed performs better as compared to the conventional method without the DT method. For instance, if the data size is around 5000 bytes, the transmission time taken by CADOM is ≈ 0.02 which is half the time taken by the conventional method without DT. We reason and attribute this behavior to the wireless channel's propagation features that are bound to create this difference between the DT received message and the information transmitted by PE_{PIOV} . Further, we also observe that when DT information in the CADOM method is available, we can see that the rate of transmission time is high initially in the first connection, due to *https* connection establishment time. The first *https* connection established is a secure connection having various steps like SSL/TLS handshake, guaranteeing encrypted data during data transmission. The handshake process requires round trips which further adds on to the time. In addition, the system may also require extra time to allocate

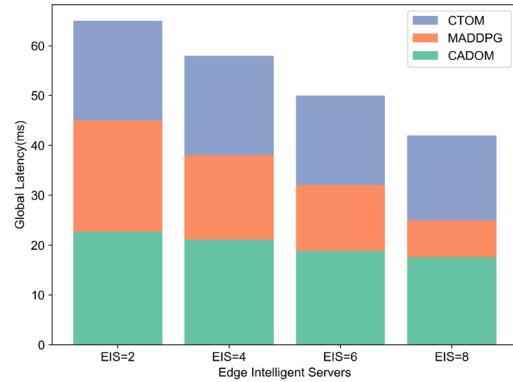
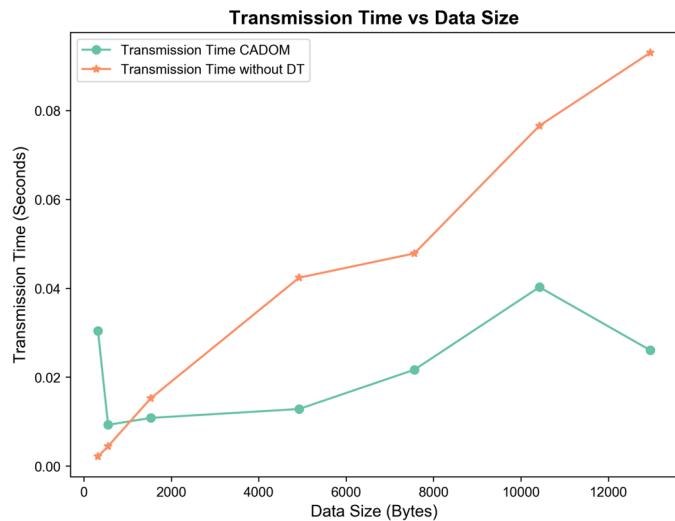


Fig. 16 The global latency of the system vs no. of SENs for different task offloading strategies

Fig. 17 Transmission time for with DT and without DT scenarios over different data sizes



resources, set up communication channels, and configure the network on the basis of the real-time data provided by the DT.

Figure 18 compares the varying data size of the input on the feedback time. Our proposed CADOM shows an initial increase in the delay of the feedback result, because of the https connection establishment time in the beginning of the simulation. Subsequently, CADOM shows a stark decline in the delay of the feedback result as compared to the without DT method.

Figure 19 compares the global latency with CADOM using DT implementation, and it shows the contributing parameters which affect the performance of the global latency. It is seen that the transmission time and feedback time both have equivalent weightage on the global latency, and the execution time has a higher weightage over the global latency performance. The global latency in our proposed system model is considerably less when compared to the global latency in Fig. 20 in without DT scenario. Figure 14

also assesses the contributing parameters affecting the performance of the global latency. The global latency in without DT scenario has strong fluctuations with higher values as compared to CADOM in Fig. 19.

In this section, we analyze the proposed reward-based mechanism using our DT simulation in MATLAB. We create a $DT_{P_{IoV}}$ model network, where the $P_{E_{IoV}}$ utilities are the vehicles, i.e., MENs and EIS-assisted RSUs, i.e., SENs. The DTs of these utilities represent the time-varying supply-demand of resources. MENs with varying preferences for SENs utilize the $DT_{P_{IoV}}$ for offloading their computational tasks to the appropriate EIS-assisted RSUs. Figure 21 illustrates how the computational resources provided by every EIS-assisted RSU in both the cases of with DT and without DT. In case of without DT of MENs, it is not possible to gather collective preference (W_n) of all MENs for all SEN_n , and the computational resources can only be purchased based on the unit cost of RSU (C_n).

Fig. 18 Feedback time for with DT and without DT scenarios over different data sizes

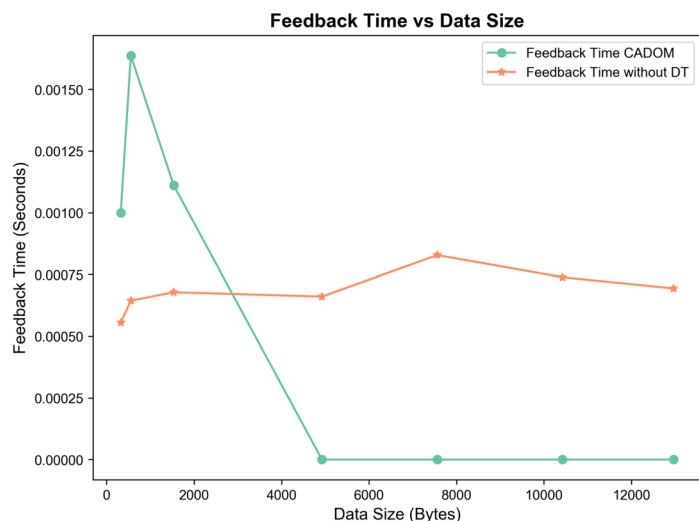
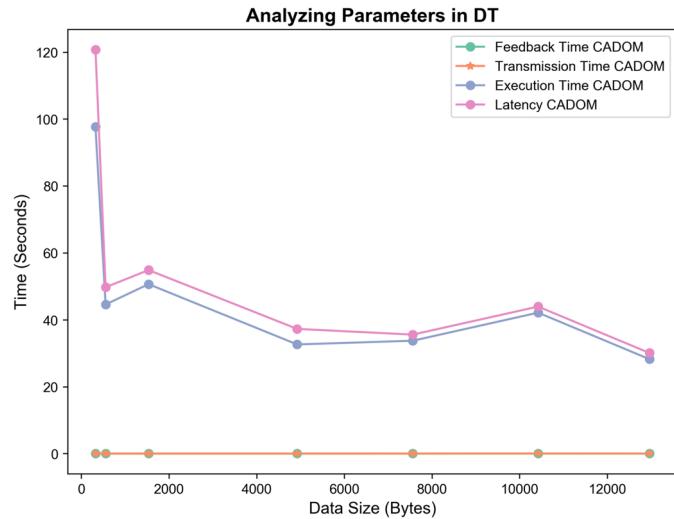


Fig. 19 Record of all parameters contributing to the latency for with DT scenario over different data sizes

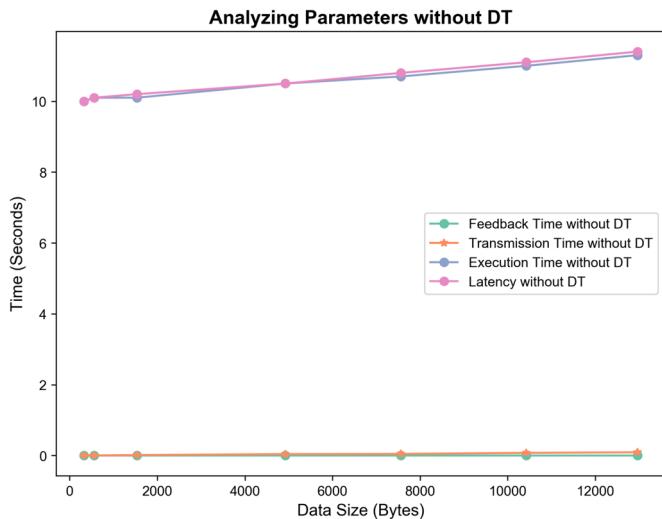


Therefore, we can say that RSU1 having lowermost unit cost delivers the maximum computing resources in without DT scenario. However, when the information of DT is accessible, the proposed reward-based mechanism considers both the unit cost of CPU in EIS-assisted RSUs and the collective vehicle preferences (W_n) for EIS-assisted RSUs for meeting the strict offloading needs of MENs. As a result, RSU1, with lower cost and relatively high collective preference (the lowermost $\frac{C_n}{W_n}$), provides the maximum number of computational resources. With the information of collective preference given by the DT, the above-discussed scheme encourages yet more suitable EIS-assisted RSUs to serve MENs as per their precise offloading needs, outperforming the approach without the DT. In Figs. 22 and 23, the two methods are compared under the influence of DT and without DT. We have considered two scenarios shown in Figs. 22 and 23, elaborated as follows: The two methods are taken

with (1) varying number of computational resources and (2) varying number of EIS-assisted RSUs, respectively.

In Fig. 22, first we taken different cases of CPU utilization of the EIS-assisted RSU, i.e., 100% usage of all EIS-assisted RSU, 95% usage of all EIS-assisted RSU, varying one of the EIS-assisted RSU to 80% and others to 95% usage, etc. The graph shows that the reward-based mechanism with DT shows a better satisfaction rate as compared to the without DT scenario. Moreover, we also tested the two cases with varying number of EIS-assisted RSUs in Fig. 23. In one case, we assume all the EIS-assisted RSUs are available and calculated its cumulative satisfaction. Further, we assumed the absence of one or two EIS-assisted RSUs in different cases and recorded its cumulative satisfaction. The vehicle satisfaction dips lower as we keep decreasing the number of EIS-assisted RSUs. However, it still performs better than without DT method.

Fig. 20 Record of all parameters contributing to the latency for without DT scenario over different data sizes



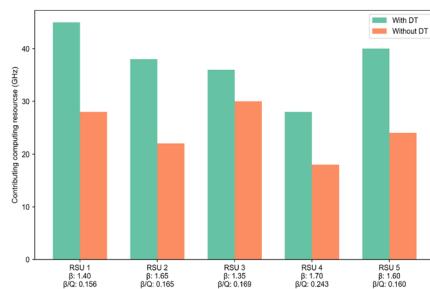


Fig. 21 Computational resources provided by every EIS-assisted RSU

Inference and Analysis

CADOM is compared with three existing methods in Table 5. The first experiment is where all tasks are executed partially in vehicle and partially in ES (PV/E). The second experiment is a random selection (RS) of the resources for the tasks. The third variation is the feedback time without DT. We witness a significant improvement in the CADOM method proposed in the paper. It displays better feedback time as compared to other methods across different data sizes.

The dataset construction part was very laborious and lengthy. The edge servers had to be connected with every other ES with different timestamps. Keeping a record of such an intricate database was one of the biggest challenges in our work. In addition, since we tested for multiple ES and vehicles, multi-simulation took a lot of time to execute considering limited computational capacity PCs in our lab. The potential of DT-empowered task offloading in IoV has played a vital role in enhancing driver's safety and reducing traffic congestion for future Intelligent Transportation Systems (ITS). The essential functionalities in IoV are generally low-latency sensing, lane merge, real-time video analytics, efficient communication, obstacle detection, etc. Moreover, the deployment of ES for wider spatial coverage also increases the network range. Considering the time-sensitive and spatial variations in the vehicular requests, there is an unnecessary resource wastage. The proposed CADOM aids in an efficient task offloading process to optimize the vehicular interactions in accordance with the

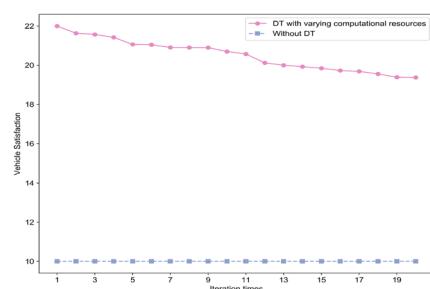


Fig. 22 Varying computational resources provided by every EIS-assisted RSU

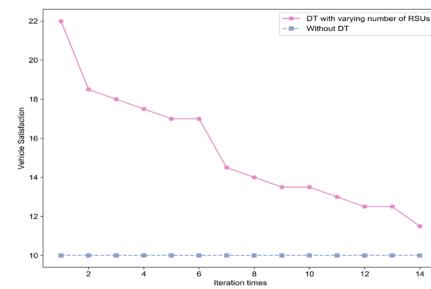


Fig. 23 Computational resources provided by varying number of EIS-assisted RSU

smart cities sustainability principles. It is important to make intelligent decisions on-road. This will avoid traffic accidents and jams, thereby reducing pollution and will establish energy-efficient PIoV. Our method emphasizes on network transmission optimization, efficient traffic management with utility maximization, latency minimization, etc. for next-generation IoTs.

Conclusion

In this article, we jointly formulate the latency-minimization and utility (vehicles and EIS-assisted RSU) service maximization problem in the PIoV scenario of smart city. To address these issues, this article experiments the utilization of DT technology by proposing a comprehensive DT-assisted CADOM for realizing the computation offloading process and a reward-based mechanism for optimizing the RA during varying supply–demand in PIoV sustainable smart city environment. For minimizing the global latency of the system, we used selection algorithm which facilitated EI-collaboration and aided in selecting suitable EIS for the offloading process. The simulation results show that the incorporation of DT has significantly reduced the response time of the system. Moreover, the latency also improves with increasing number of EIS in CADOM as compared to the existing conventional offloading methods.

Table 5 Comparative analysis of the existing work with CADOM

S. no	Size (kB) (Client sending files)	PV/E	RS	CADOM CADOM	Feedback time(s) w/o DT
S. no	PV/E	RS	CADOM		
1	324	1.9	2.2	0.001	0.003
2	555	2.1	2.9	0.002	0.002
3	1535	2.5	3.5	0.001	0.002
4	4925	2.6	4.4	0	0.002
5	10,423	3.1	6.1	0	0.001
6	12,961	4.2	6.99	0	0.001
7	7562	3.5	5.8	0	0.001

Further, the simulation results of the proposed reward-based mechanism show optimal EIS-assisted RSU allocation to the MENs by considering both the MEN preferences and the EIS-assisted RSU's CPU unit cost to meet the requirements of the dynamic task offloading process. Our extensive simulations performed show the feasibility of the proposed method which results in improved overall response time and improved resource contribution of EIS-assisted RSU to enhance the service quality. The proposed CADOM aids in an efficient task offloading process to optimize the vehicular interactions in accordance with the smart cities sustainability principles. It is important to make intelligent decisions on-road. This will avoid traffic accidents and jams, thereby reducing pollution and will establish energy-efficient PIoV. Our method emphasizes on network transmission optimization, efficient traffic management with utility maximization, latency minimization, etc. for next-generation IoVs.

Acknowledgements The authors would like to thank the anonymous reviewer(s) for their invaluable comments and feedback. This work is carried out in the Disruptive Technologies lab, which is supported by the Department of Science and Technology (DST), Govt. of India, in the form of FIST Level-1 grant to the Department of CSIS, Birla Institute of Technology and Science Pilani.

Author Contributions Conceptualization: Oshin Rawlley and Shashank Gupta; methodology: Oshin Rawlley, Jyotsana Chandrakar, Manisha K Johnson, Chahat Kalra; formal analysis and investigation: Oshin Rawlley and Shashank Gupta; writing—Oshin Rawlley; writing—review and editing: Oshin Rawlley, Jyotsana Chandrakar, Manisha K Johnson, Chahat Kalra; funding acquisition: Shashank Gupta; resources: Shashank Gupta; supervision: Shashank Gupta

Funding This work is also supported by CHANAKYA Fellowships of IITI DRISHTI CPS Foundation under the National Mission on Interdisciplinary Cyber-Physical System (NM-ICPS) of the Department of Science and Technology, Government of India. Grant no.: CF-2022-PhD-000002. This work is also supported by I-DAPT HUB FOUNDATION, IIT (BHU), Varanasi for the project ref. no. I-DAPT/IIT (BHU)/2023-24/ Project Sanction/44. dated 19-09-2023.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of Interest The authors declare no competing interests.

References

- Liu C, Ke L. Cloud assisted internet of things intelligent transportation system and the traffic control system in the smart city. *J Control Decision*. 2023;10(2):174–87.
- Krishankumar R, Ecer F. Selection of IoT service provider for sustainable transport using q-rung orthopair fuzzy CRADIS and unknown weights. *Appl Soft Comput*. 2023;132:109870.
- Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of things for smart cities. *IEEE Internet Things J*. 2014;1(1):22–32.
- Ang LM, Seng KP, Zungeru AM, Ijemaru JK. Big sensor data systems for smart cities. *IEEE Internet Things J*. 2017;4(5):1259–71.
- Csáji BC, Kemény Z, Pedone G, Kuti A, Vánca J. Wireless multi-sensor networks for smart cities: a prototype system with statistical data analysis. *IEEE Sens J*. 2017;17(23):7667–76.
- Musa S. Smart cities—a road map for development. *IEEE Potentials*. 2018;37(2):19–23.
- Ang L-M, Seng KP, Ijemaru GK, Zungeru AM. Deployment of IoV for smart cities: applications, architecture, and challenges. *IEEE Access*. 2018;7:6473–92.
- Liu N. Internet of vehicles: your next connection. *Huawei WinWin*. 2011;11:23–8.
- Golestan K, Soua R, Karray F, Kamel MS. Situation awareness within the context of connected cars: a comprehensive review and recent trends. *Inf Fus*. 2016;29:68–83.
- Wan J, Zhang D, Zhao S, Yang LT, Lloret J. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Commun Mag*. 2014;52(8):106–13.
- Gandotra P, Jha RK, Jain S. A survey on device-to-device (d2d) communication: Architecture and security issues. *J Netw Comput Appl*. 2017;78:9–2.
- Bonomi F, et al. The smart and connected vehicle and the internet of things. In: Invited talk, workshop on synchronization in telecommunication systems. sn, 2013.
- Gehrman C, Gunnarsson M. A digital twin based industrial automation and control system security architecture. *IEEE Trans Ind Inf*. 2019;16(1):669–80.
- Wang D, Zhang Z, Zhang M, Meixia F, Li J, Cai S, Zhang C, Chen X. The role of digital twin in optical communication: fault management, hardware configuration, and transmission simulation. *IEEE Commun Mag*. 2021;59(1):133–9.
- Jiang L, Zheng H, Tian H, Xie S, Zhang Y. Cooperative federated learning and model update verification in blockchain-empowered digital twin edge networks. *IEEE Internet Things J*. 2021;9(13):11154–67.
- Al-Shatri H, Müller S, Klein A. Distributed algorithm for energy efficient multi-hop computation offloading. In: 2016 IEEE international conference on communications (ICC), pp 1–6. IEEE, 2016.
- Xu X, Shen B, Ding S, Srivastava G, Bilal M, Khosravi MR, Menon VG, Jan MA, Wang M. Service offloading with deep q-network for digital twinning-empowered internet of vehicles in edge computing. *IEEE Trans Ind Inf*. 2020;18(2):1414–23.
- Yang L, Yao H, Wang J, Jiang C, Benslimane A, Liu Y. Multi-UAV-enabled load-balance mobile-edge computing for IoT networks. *IEEE Internet of Things J*. 2020;7(8):6898–908.
- Zhang J, Guo H, Liu J, Zhang Y. Task offloading in vehicular edge computing networks: a load-balancing solution. *IEEE Trans Veh Technol*. 2019;69(2):2092–104.
- Zeng F, Chen Q, Meng L, Jinsong W. Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing. *IEEE Trans Intell Trans Syst*. 2020;22(6):3247–57.
- Eichelberger AH, McCarr AT. Toyota drivers' experiences with dynamic radar cruise control, pre-collision system, and lane-keeping assist. *J Saf Res*. 2016;56:67–73.
- Naus GJ, Vugts RP, Ploeg J, van De Molengraft MJ, Steinbuch M. String-stable CACC design and experimental validation: a frequency-domain approach. *IEEE Trans Veh Technol*. 2010;59(9):4268–79.
- Shi J, Jun D, Wang J, Wang J, Yuan J. Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning. *EEE Trans Veh Technol*. 2020;69(12):16067–81.
- Sun W, Wang P, Ning X, Wang G, Zhang Y. Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles. *IEEE Internet Things J*. 2021;9(8):5839–52.

25. Chunhua H, Fan W, Zeng E, Hang Z, Wang F, Qi L, Bhuiyan MZA. Digital twin-assisted real-time traffic data prediction method for 5G-enabled internet of vehicles. *IEEE Trans Ind Inf.* 2021;18(4):2811–9.
26. Wang D, Song B, Lin P, Yu FR, Du X, Guizani M. Resource management for edge intelligence (EI)-assisted IoV using quantum-inspired reinforcement learning. *IEEE Internet of Things Journal.* 2021;9(14):12588–600.
27. Liu T, Tang L, Wang W, He X, Chen Q, Zeng X, Jiang H. Resource allocation in DT-assisted internet of vehicles via edge intelligent cooperation. *IEEE Internet Things J.* 2022;9(18):17608–26.
28. Talebkhah M, Sali A, Khodamoradi V, Khodadadi T, Gordan M. Task offloading for edge-IoV networks in the industry 4.0 era and beyond: a high-level view. *Eng Sci Technol Int J.* 2024;54:101699.
29. Gonzalez-Martín M, Sepulcre M, Molina-Masegosa R, Gozalvez J. Analytical models of the performance of C-V2X mode 4 vehicular communications. *IEEE Trans Veh Technol.* 2018;68(2):1155–66.
30. Hou X, Ren Z, Wang J, Cheng W, Ren Y, Chen K-C, Zhang H. Reliable computation offloading for edge-computing-enabled software-defined IoV. *IEEE Internet Things J.* 2020;7(8):7097–111.
31. Gong Y, Wei Y, Feng Z, Yu FR, Zhang Y. Resource allocation for integrated sensing and communication in digital twin enabled internet of vehicles. *IEEE Trans Veh Technol.* 2022.
32. Yueyue DD, Maharjan S, Qiao G, Zhang Y. Artificial intelligence empowered edge computing and caching for internet of vehicles. *IEEE Wirel Commun.* 2019;26(3):12–8.
33. Zhang K, Leng S, Peng X, Pan L, Maharjan S, Zhang Y. Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE Internet Things J.* 2018;6(2):1987–97.
34. Guo J, Luo W, Song B, Yu FR, Du X. Intelligence-sharing vehicular networks with mobile edge computing and spatiotemporal knowledge transfer. *IEEE Netw.* 2020;34(4):256–62.
35. Feng J, Yu FR, Pei Q, Chu X, Du J, Zhu L. Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach. *IEEE Internet Things J.* 2019;7(7):6214–28.
36. Islam S, Badsha S, Sengupta S, La H, Khalil I, Atiquzzaman M. Blockchain-enabled intelligent vehicular edge computing. *IEEE Netw.* 2021;35(3):125–31.
37. Alharthi M, Taha A-EM, Hassanein HS. An architecture for software defined drone networks. In: ICC 2019-2019 IEEE international conference on communications (ICC), pp 1–5. IEEE, 2019.
38. Yunlong L, Huang X, Zhang K, Maharjan S, Zhang Y. Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks. *IEEE Trans Ind Inf.* 2020;17(7):5098–107.
39. Sun W, Zhang H, Wang R, Zhang Y. Reducing offloading latency for digital twin edge networks in 6G. *IEEE Trans Veh Technol.* 2020;69(10):12240–51.
40. LiWang M, Dai S, Gao Z, Du X, Guizani M, Dai H. A computation offloading incentive mechanism with delay and cost constraints under 5G satellite-ground IoV architecture. *IEEE Wirel Commun.* 2019;26(4):124–32.
41. Deng T, Chen Y, Chen G, Yang M, Du L. Task offloading based on edge collaboration in MEC-enabled IoV networks. *Journal of Communications and Networks.* 2023.
42. Awada U, Zhang J, Chen S, Li S, Yang S. Resource-aware multi-task offloading and dependency-aware scheduling for integrated edge-enabled IoV. *J Syst Archit.* 2023;141:102923.
43. Wang H, Li X, Ji H, Zhang H. Federated offloading scheme to minimize latency in MEC-enabled vehicular networks. In: 2018 IEEE globecom workshops (GC Wkshps), pp 1–6. IEEE, 2018.
44. Wang Y, Hu X, Guo L, Yao Z. Research on V2I/V2V hybrid multi-hop edge computing offloading algorithm in IoV environment. In: 2020 IEEE 5th international conference on intelligent transportation engineering (ICITE), pp 336–340. IEEE, 2020.
45. Hou X, Li Y, Chen M, Di W, Jin D, Chen S. Vehicular fog computing: a viewpoint of vehicles as the infrastructures. *IEEE Trans Veh Technol.* 2016;65(6):3860–73.
46. Chen X, Jiao L, Li W, Xiaoming F. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans Netw.* 2015;24(5):2795–808.
47. Zhuang W, Ye Q, Lyu F, Cheng N, Ren J. SDN/NFV-empowered future IoV with enhanced communication, computing, and caching. *Proc IEEE.* 2019;108(2):274–91.
48. Shen B, Xiaolong X, Qi L, Zhang X, Srivastava G. Dynamic server placement in edge computing toward internet of vehicles. *Comput Commun.* 2021;178:114–23.
49. Zhao J, Sun X, Li Q, Ma X. Edge caching and computation management for real-time internet of vehicles: an online and distributed approach. *IEEE Trans Intell Trans Syst.* 2020;22(4):2183–97.
50. Zhao L, Li H, Lin N, Lin M, Fan C, Shi J. Intelligent content caching strategy in autonomous driving toward 6G. *IEEE Trans Intell Trans Syst.* 2021;23(7):9786–96.
51. Umber SY, Liu SJ, Li Y, Jiang T. Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing. *IEEE Trans Wirel Commun.* 2020;20(1):360–74.
52. Lin K, Li Y, Zhang Q, Fortino G. AI-driven collaborative resource allocation for task execution in 6G-enabled massive IoT. *IEEE Internet Things J.* 2021;8(7):5264–73.
53. Qiao G, Leng S, Maharjan S, Zhang Y, Ansari N. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks. *IEEE Internet Things J.* 2019;7(1):247–57.
54. Alfakih T, Hassan MM, Gumaei A, Savaglio C, Fortino G. Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA. *IEEE Access.* 2020;8:54074–84.
55. Xu X, Jiang Q, Zhang P, Cao X, Khosravi MR, Alex LT, Qi L, Dou W. Game theory for distributed IoV task offloading with fuzzy neural network in edge computing. *IEEE Trans Fuzzy Syst.* 2022;30(11):4593–604.
56. Zhou J, Wu F, Zhang K, Mao Y, Leng S. Joint optimization of offloading and resource allocation in vehicular networks with mobile edge computing. In: 2018 10th International conference on wireless communications and signal processing (WCSP), pp 1–6. IEEE, 2018.
57. Biesinger F, Kraß B, Weyrich M. A survey on the necessity for a digital twin of production in the automotive industry. In: 2019 23rd International conference on mechatronics technology (ICMT), pp 1–8. IEEE, 2019.
58. Liu J, Dong Y, Liu Y, Li P, Liu S, Wang T. Prediction study of the heavy vehicle driving state based on digital twin model. In: 2021 IEEE international conference on power electronics, computer applications (ICPECA), pp 789–797. IEEE, 2021.
59. Lee A, Kim J, Jang I. Movable dynamic data detection and visualization for digital twin city. In: 2020 IEEE international conference on consumer electronics-asia (ICCE-Asia), pp 1–2. IEEE, 2020.
60. Wang Z, Liao X, Zhao X, Han K, Tiwari P, Barth MJ, Wu G. A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems. In: 2020 IEEE 91st vehicular technology conference (VTC2020-Spring), pp 1–6. IEEE, 2020.
61. Liu Y, Wang Z, Han K, Shou Z, Tiwari P, Hansen JHL. Sensor fusion of camera and cloud digital twin information for intelligent vehicles. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp 182–187. IEEE, 2020.
62. Zhou Z, Liu P, Feng J, Zhang Y, Mumtaz S, Rodriguez J. Computation resource allocation and task assignment optimization in vehicular fog computing: a contract-matching approach. *IEEE Trans Veh Technol.* 2019;68(4):3113–25.

63. Zheng Z, Song L, Han Z, Li GY, Poor HV. A Stackelberg game approach to large-scale edge caching. In: 2018 IEEE global communications conference (GLOBECOM), pp 1–6. IEEE, 2018.
64. Wang C, Liang C, Yu FR, Chen Q, Tang L. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Trans Wirel Commun.* 2017;16(8):4924–38.
65. Liang C, Yu FR. Distributed resource allocation in virtualized wireless cellular networks based on ADMM. In: 2015 IEEE conference on computer communications workshops (INFOCOM WKSHPS), pp 360–365. IEEE, 2015.
66. Liu K, Dai P, Lee VC, Ng JK-Y, Son SH. Distributed task offloading and workload balancing in IoV. In: Toward connected, cooperative and intelligent IoV: frontier technologies and applications, pp 173–195. Springer, 2023.
67. Li Y, Li L, Xia Y, Zhang D, Wang Y. Multi-leader single-follower Stackelberg game task offloading and resource allocation based on selection optimization in internet of vehicles. *IEEE Access*, 2023.
68. Zheng Z, Song L, Han Z. Bridge the gap between ADMM and Stackelberg game: incentive mechanism design for big data networks. *IEEE Signal Process Lett.* 2017;24(2):191–5.
69. Yang D, Xue G, Fang X, Tang J. Incentive mechanisms for crowdsensing: crowdsourcing with smartphones. *IEEE/ACM Trans Netw.* 2015;24(3):1732–44.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.