ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SOLVING DYNAMIC TASK OFFLOADING IN VEHICULAR FOG COMPUTING USING PARTIALLY OBSERVABLE MARKOV DECISION PROCESS BASED DEEP DETERMINISTIC POLICY GRADIENT APPROACH

J. SHANKAR[1], S. R. NAGARAJA[2]

[1,2]School of Computer Science, Presidency University, Bengaluru, India

E-mail: [1]shankar.j@presidencyuniversity.in, [2]nagarajasr@presidencyuniversity.in

ABSTRACT

Vehicular Fog Computing (VFC) becomes the potential solution for mitigating the vehicular computation load. In graded Vehicular Fog Computing (VFC), vehicles function as mobile fog nodes at network edge, delivering consistent as well as minimum-latency services. Due to limited onboard computational resources, vehicles offload intensive tasks to nearby Roadside Units (RSU). To minimize a computational weight at RSUs, VFC is utilized for the computational-exhaustive tasks. Within this framework, vehicles serve as part of the infrastructure, facilitating communication, monitoring, and resource sharing among fog nodes. This makes efficient resource allocation a critical factor for overall system performance. Thus, this research proposes the Partially Observable Markov Decision Process-assisted Deep Deterministic Policy Gradient (POMDP-DDPG) approach for an effective task offloading in VFC. The proposed approach is a Reinforcement Learning (RL) that utilises the global data to identify a better connection between the RSU and the fog servers. The experimental results specify that the proposed POMDP-DDPG approach attains better results as compared to the existing approaches

Keywords: *Deep Deterministic Policy Gradient, Partially Observable Markov Decision Process, Reinforcement Learning, Roadside Units, Vehicular Fog Computing*

1. INTRODUCTION

Through the development of the Internet of Vehicles (IoV) and autonomous driving technologies, a broad variety of vehicular areas has emerged to improve driving knowledge. Applications like crowd-sensing, augmented reality, and intellectual navigation are often computationally intensive and delay-sensitive, requiring efficient processing and real-time responsiveness [1][2]. To encounter an enhanced computational mandate of vehicles, the Vehicular Fog Computing (VFC) is used for communicating an optimal computing resource between cooperatively [3]. Compared to relying on centralized cloud services, Vehicular Fog Computing (VFC) offers faster and more responsive computing services to vehicles by significantly reducing latency [4]. Additionally, VFC helps alleviate the computational load on edge servers located at base stations (BS) or roadside units (RSUs), ensuring more efficient task handling and resource utilization [5]. Hence, it is important for developing a computation offloading strategy in VFC that enhances computational ability significantly [6][7]. Nowadays, the research on VFC has obtained substantial courtesies, and different vehicular computation offloading mechanisms are developed for improving an usage of VFC. Cloud computing is observed as an important solution to the above-discussed problems in vehicular networks [8][9]. In cloud-assisted vehicular networks, tasks are offloaded because of their greater computation competences of cloud Virtual Machines (VMs) [10].

Nevertheless, a presence of a corporal break among vehicles and the cloud server leads to an important latency breach, that minimizes an effectiveness as well as significance of task offloading [11]. Hence, decentralized framework with greater necessities is important to minimise experienced latency and efficiently manage time and latency tasks, complementing the capabilities of cloud computing [12] [13]. To significantly attain computation tasks offloading of mobile devices to edge servers, the count of exertions has been introduced. Deep Reinforcement Learning (DRL) is a subdivision of Artificial Intelligence (AI), that uses a visual capability of Deep Learning (DL) and a policymaking ability of Reinforcement Learning (RL) [14][15]. DRL attains a ideal offloading mechanism through directly cooperating with the dynamic vehicular network. Nevertheless, traditional RL approaches face challenges due to the "curse of dimensionality" [16]. To address this, integrating DL with RL, resulting in DRL, improves the model's ability to tackle complex optimization problems in dynamic and unpredictable environments [17]. In recent times, various studies have accomplished the policy-assisted DRL approaches for task scheduling in vehicular settings [18] [19]. From the above discussed studies, researchers enhance the performance indices such as reducing delay, latency and energy consumption. More researchers have concentrated on enhancing an individual performance metric to improve the model's performance. Only some researchers have optimized various performance indices while seeing the priority and deadline of input tasks [20]. In VFC, vehicles play as an infrastructure, allowing the fog nodes to communicate, as well as resource sharing, making resource arrangement a critical aspect of system efficiency. Thus, this research proposes the Partially Observable Markov Decision Process-based Deep Deterministic Policy Gradient (POMDP-DDPG) approach for an effective task offloading in VFC.

The key notations of this research are as trails:

- This research introduces an approach for a vehicular application of task offloading issues through combining task dependencies, supportive computation in VFC as well as vehicle mobility.
- This research models the reliable task offloading issue as an MDP as well as introduces a new approach according to the DDPG through experience delay and target network.
- This research performs simulation results to evaluate an effectiveness of an introduced task offloading method in reducing delay as well as energy consumption, surpassing a significance of the existing approaches.

This paper is organised as trails: Section 2 offerings the literature survey. Section 3 gives the proposed methodology. Section 4 illustrates the experimental results and Section 5 provides the conclusion.

2. LITERATURE SURVEY

Due to the rapid advancement of vehicular technologies within the Internet of Vehicles (IoV), a growing amonut of studies are exploring an idea of task offloading in VFC.

Kaushik Mishra et al. [21] implemented the determined the task offloading methods in VFC models as well as developed a FL-assisted Deep Q-Learning-assisted (FedDQL) method for ideal task offloading in the cooperative computing standard. An introduced offloading approach employs computations, communications, and offloading, as well as resource utilization, considering latency as well as energy consumption. A trade-off among latency as well as computing limitations was considered for model estimation. A FedDQL approach was estimated for task-reliant sets for evaluation of the approach's effectiveness. However, an implemented method focused on federated learning but lacked a consideration for task priority and did not optimise for highly dynamic mobility or network variance.

Vivek Sethi and Sujata Pal [22] presented the online FDL-assisted Offloading approach for Vehicular fog computing (FedDOVe) that combinedly optimised an energy consumption over RSUs as well as load balancing over fog servers. FedDOVe was a RL method which utilized global data for the determination of ideal integration among RSUs and fog servers. However, the presented method was efficient only for RSU-fog load balancing, but it overlooks task-level granularity and task dependency modelling.

Bushra Jamil et al. [23] implemented the Proximal Policy Optimization (PPO)-assisted Intelligent, priority as well as deadline-aware online as well as distributed Resource Allocation and Task Scheduling approach, namely IRATS, in CFC. IRATS estimated resource allocation issues as a MDP for a reduction of task waiting time as well as delay. The author designed a task allocator for sequential estimation of offloaded tasks based on their priorities by multi-level queues. However, the implemented method was constrained through static task assumptions and doesn't integrate vehicular mobility modelling.

Zhiwei Wei et al. [24] presented the many-to-many task offloading paradigm with respect to the vehicular trading network. The presented paradigm allowed a computational resource interchange over various VFC subsystems and chose a multi-tier task offloading outcomes in terms of trading consensus. The offloading and service cooperation process is modelled as a POMDP, and a Multi-Agent Gated Actor Attention Critic (MA-GAC) method was introduced to enable appropriate and consistent task offloading as well as collaboration between vehicles. Nevertheless, the presented method introduced high communication overhead and lacked scalability.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Ning Chen et al. [25] developed the spectral graph theory-based resource orchestration approach through integration of Virtual Network Embedding (VNE) as well as DRL. A multi-layer mechanism based on Graph Convolutional Networks (GCNs) was specifically developed to compute node embedding probabilities. In this approach, fog nodes effectively captured spatial structure information through integration of their features with those of neighbouring nodes. This integration supports addressing the limitations of conventional heuristic-based Virtual Network Embedding (VNE) methods. Furthermore, fog link embedding was employed through breadth-first search (BFS). Eventually, the efficiency of the developed mechanism was methodically and thoroughly demonstrated concerning long-term regular revenue, average revenue-cost ratio, as well as VNR receiving rate by simulation scenarios. However, the developed model was insufficient in modelling vehicular task priorities and execution constraints.

Ahmad Naseem Alvi et al. [26] designed a utility function for each offloaded vehicular task by taking into account both the task's priority and its association duration with the fog node. Moreover, a knapsack-assisted task scheduling method was developed to ideally perform task offloading. A primary concept of this approach was to ideally implement more time-consuming tasks through a fog computing node for rapid execution and minimum penetrating tasks in the cloud. However, the designed approach utilised heuristic rules without adaptive learning for dynamic environments.

Mekala Ratna Raju et al. [27] presented a mobility-aware task scheduling issue to combinedly optimize service latency of the tasks and energy consumption of vehicles. Primarily, the Markov Renewal Process (MRP) was performed to capture the vehicle's mobility. Following that, it established that an introduced fitness function was a drone submodular function as well as offered a greedy heuristic for the enhancement of the submodular function through an approximation. Afterwards, a hybrid optimisation approach of Particle Swarm Optimization (PSO) and Simulated Annealing (SA) was developed to address mobility-aware task scheduling issues for large-scale networks. However, heuristic-based approaches like PSO-SA may converge prematurely in complex decision spaces.

Shanchen Pang et al. [28] implemented DRL approach to improve a task scheduling mechanism ensure real-time and efficient task execution. Particularly, task dependencies were modelled through a Directed Acyclic Graph (DAG) as well as dynamically adjusted the weights associated with delay as well as energy consumption during task offloading. By addressing the challenges of task offloading in vehicle edge cloud setting and incorporating task needs into the MDP, the approach enabled a more structured and effective problem-solving strategy. To acquire rapid ideal offloading decisions, a Double Deep Q-Network (DDQN) was employed through mobility management strategies. A penalty strategy was developed into DDQN to execute penalties at vehicle application exceeds its execution deadline.

Although existing studies have significantly enhanced task offloading in VFC, most of them remain constrained to either latency minimisation or energy-aware optimisation. FedDQL [21] and FedDOVe [22] provide effective learning-based solutions, but do not effectively consider task priority, dependency, and the dynamic mobility of vehicles. Likely, IRATS [23] integrated deadline and priority constraints, but it assumes relatively consistent execution conditions and does not solve rapidly differing vehicular topology. Multi-agent method named MA-GAC [24] was introduced the considerable communication overhead and complex to scale in dense vehicular environments. Graph-based resource orchestration method [25] effectively model resource relationships, but do not model the urgency and execution constraints of individual tasks. Moreover, the heuristic and metaheuristic approaches [26]-[28] often suffer from premature convergence, static assumptions, and constrained adaptability when network conditions change in real time.

Hence, an effective task offloading framework in VFC should simultaneously support multi-objective optimisation, partial system observability, vehicle mobility, task dependency, and task priority. Hence, this research proposes the POMDP-DDPG framework to address these shortcomings through integrating the decision-making capability of DDPG with indecision modelling of POMDP.

2.1 Research Gap

Despite extensive research in Vehicular Fog Computing (VFC), existing approaches often focus on optimizing a single metric, assume complete system observability, or fail to adapt to dynamic environments. Moreover, they rarely consider task priority, deadline constraints, and inter-task dependencies simultaneously. The proposed POMDP-DDPG model overcomes these gaps by performing multi-objective optimisation under partial observability, integrating priority-aware and dependency-aware scheduling. Leveraging reinforcement learning, it dynamically adapts to changing network conditions, ensuring efficient and reliable task offloading in real-time VFC scenarios.

3. PROPOSED METHODOLOGY

Primarily, this portion implements a VFC approach for task offloading. Then, various system frameworks such as vehicular task as well as computation models are described. Moreover, a problem formulation through primary aims is described. Figure 1 demonstrates an architecture of the VFC network.

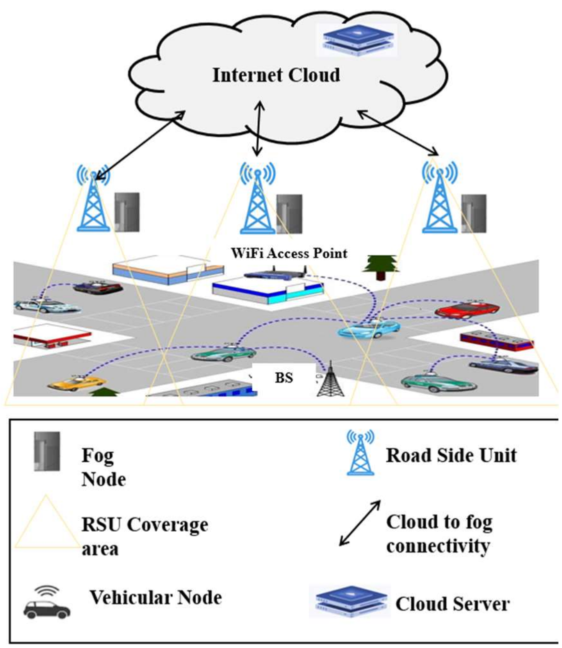

Figure 1: Architecture of the VFC network

3.1 Framework of Vehicular Fog Computing

VFC's decentralized framework involves multiple layers such as vehicular, Fog and cloud. The comprehensive details of these layers are provided in the subsequent.

3.1.1 Cloud Layer

This layer operates at highest level of the architecture and comprises centralised cloud infrastructure connected through a core network. It is equipped with ample computing and storage resources. Data collected from vehicles is transmitted to cloud using advanced communication advancements such as Long-Term Evolution (LTE), 4G, and 5G. This data is used for caching, performing long-term analytics, and aiding in strategic decision-making. The cloud conducts advanced data analysis to identify trends, make informed decisions, and reveal hidden patterns. Additionally, the cloud hosts an Adaptive Signal Controller module that receives data and outcomes from upper-tier fog nodes for further analysis, visualisation, and storage.

3.1.2 Fog Layer

This is a centre layer among cloud as well as vehicle layers, which comprises a network of geographically distributed and heterogeneous fog nodes. These nodes manage the aggregation, real-time processing, and temporary storage of data generated through connected vehicles and infrastructure. Examples of devices in this layer include Roadside Units (RSUs), wireless access points, and base stations (BSs). Communication between vehicles and fog nodes occurs via short-range wireless technologies, such as IEEE 802.11p and Dedicated Short-Range Communication (DSRC). Tasks generated with vehicles are first gathered through IoV Data Collector module deployed on lower-level fog devices. These modules forward a processed data and non-urgent tasks to higher-level fog nodes for additional computation and analysis.

3.1.3 Vehicular Layer

This layer consists of interconnected vehicles that possess capabilities for communication, service exchange, and local data storage. Vehicles collect data from a range of sources, including onboard sensors, intra-vehicle communication systems (V2V), vehicle-to-infrastructure (V2I) interfaces, environmental sensors, and equipment like infrared, ultrasonic, microwave as well as speed cameras. A gathered data is then forwarded to the fog layer through wireless communication technologies. Few vehicles in this layer having limited computing and storage capabilities which have the capability to offload their computational tasks to adjacent vehicles that contains available resources. These vehicles produce various types of tasks. Tasks that require more processing power are transmitted to edge-level fog nodes named RSUs or base stations for execution.

3.2 System Model

The disseminated vehicular network operating on a one-way road are considered, which comprises a single RSU and N vehicles. The system timeline is segmented into discrete time slots, during which the system state remains unchanged but may evolve across multiple slots. The RSU possesses computational capabilities to execute tasks and manage resource allocation, maintaining comprehensive information about all available computing resources. Its communication range surpasses that of individual vehicles, allowing it to monitor the positions, velocities, and computational capacities of all vehicles within range. In this model, vehicles are categorised into two distinct types:

- Task Vehicle: These vehicles contain inappropriate computational capacity, hence, needs to offload the computation tasks.
- Service Vehicle: These vehicles contain optimal computational resources, hence, receive offload desires from different vehicles.

This research considers that a particular time T, there are M task vehicles represented as $M = V_{t1}, V_{t2}, V_{t3}, \dots, V_{tn}$ and N service vehicles denoted as $N = V_{s1}, V_{s2}, V_{s3}, \dots, V_{sn}$. If a task vehicle agrees for offloading a task, it primarily transmits a request to RSU. Afterwards, it implements a negotiator to choose a relevant host in a communication extent of task-generated vehicle. The service is chosen as host fog node. Then, a task is transmitted to a service vehicle. The task vehicle produces the group of tasks $T = \{t_1, t_2, t_3, \dots, t_n\}$ and its outline is $\{C_n, P_n, \tau_n\}$, here, C_n denotes a calculated obligation such as count of CPU cycles required to implement a task is represented by C_n, P_n specifies the task priority and τ_n denotes a maximum tolerable delay.

3.2.1 Task Model

This study classifies vehicular applications into four categories based on their timing sensitivity. Complex applications require strict adherence to firm limits as well as should be processed into a defined time limit. Near Real-Time applications, while slightly more tolerant to delays than hard real-time systems, still demand rapid execution within constrained timeframes and are thus assigned the next importance in task scheduling. Soft Real-Time requests endure minor delays without significant performance degradation and are therefore ranked third in the priority hierarchy. Lastly, Delay-Tolerant applications can withstand more substantial delays and are given the lowest priority. To manage these varying priorities, a multi-level queue scheduling algorithm is employed. Incoming tasks are assigned to one of the four queues based on their application type. The scheduler gives precedence to tasks in higher-priority queues, ensuring they are executed before those in lower-priority queues. If higher-level queues are empty, the scheduler processes tasks from the next available queue. As the execution model is non-preemptive, a task currently in progress runs to completion before a new task is selected for execution.

3.2.2 Compute model

According to the introduced VFC framework, a task has to be implemented on RSU or the service vehicle. A task's endwise delay is based on data transmission tT_s , waiting time wT_s, as well as execution time $exTime_s$, which are formulated in Equations (1) to (4) as follows:

$$TD_{etoe} = exTime_s + wT_s + tT_s \tag{1}$$

$$tT_s = T_{cd} + T_{qd} + T_{td} + T_{pd} \tag{2}$$

$$exTime_s = \frac{C_n}{Mips_{hos}} \tag{3}$$

$$wT_s = \frac{\sum_{i=1}^{P_n} C_n \; of \; all \; tasks \; in \; Q_i}{Mips_{host}} \tag{4}$$

Here, TD_{etoe} specifies the task's end-to-end delay; T_{cd}, T_{qd}, T_{td} and T_{pd} demonstrates the delays of processing, queuing, transmission as well as propagation. C_n means the compute size of a task; $Mips_{host}$ represents an obtainable MIPS of host fog device. P_n denotes the priority of a task. A task's waiting time is estimated through total of calculated lengths of each task in the upper queues, as well as tasks in the applied queue if present, task partitioned through obtainable MIPS of host fog node.

3.2.3 Mobility model

In this scenario, multiple service vehicles fall within a communication array of a task-generating vehicle, and all of them are also under a coverage area of the RSU. These service vehicles are mobile, constantly changing their positions and moving at varying speeds. The selection of suitable host vehicles for task execution depends on several dynamic factors, including their current distance from the task vehicle, velocity, and the overall traffic density in the area. Enhanced traffic density generally results in a reduced average distance between vehicles. However, due to the continuous mobility and varying speeds of vehicles within the communication range, not all vehicles are suitable candidates for hosting and executing offloaded tasks.

3.3 Task Offloading using POMDP-DDPG

The DDPG approach is introduced in this research for the identification and understanding of an optimal multi-objective control policy in task offloading. As compared to other model-assisted control approaches, which are deeply based on precise system modelling. However, a DDPG approach is most appropriate for the sequential indeterminate optimal control issues. A DDPG agent learns from long-term rewards by utilizing binary opponent value function depictions for various targets: an opponent function guides a decision-making of an actor, while also helping to refine and update the control policy based on lower value function estimates [29].

The DDPG approach involves three important networks such as actor, critic as well as understanding repetition buffer. An actor network is utilized for mapping the states to maintain action. Critic networks are performed for validating state value and state-action pairs state-action pairs. An experience replay buffer is responsible for storing past interactions (transitions) to improve learning efficiency. To enhance the stability of the DDPG approach, two sets of networks such as the target actor as well as critic are performed to estimate target values. Moreover, the replay buffer stores a greater count of transitions as well as arbitrary samples small batches during training. This random sampling helps break the correlation between consecutive transitions, leading to more stable and effective updates. An actor network is parameterized through θ^{μ} which a result of $a(t) = \mu(s(t);\theta^{\mu})$, whereas a result of a target actor network is $a'(t) = \mu'(s(t+1);\theta^{\mu'})$. A result of critic network is $q = Q(s(t), a(t);\theta^{q})$ and an output of target critic network is $q' = Q'(s(t), a(t)';\theta^{q'})$.

At POMDP training procedure, the actor network's parameters are updated through a gradient descent approach in a direction of minimizing a loss, which is formulated in Equation (5) as follows:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{m}\sum_{i=1}^{m}[\nabla_a Q(s,a;\theta^{q})|s_i, \mu(s_i)\nabla_{\theta^{\mu}}\mu(s;\theta^{\mu})|s_i] \quad (5)$$

Here, m specifies a batch size. Parameter θ^{q} of a critic network are modernized through reducing a subsequent loss function, which is formulated in Equations (6) and (7) as:

$$L = \frac{1}{m}\sum_{i=1}^{m}[y_i - Q(s_i, a_i;\theta^{q})]^2 \quad (6)$$

$$y_i = r_i + \gamma Q'\left(s_{i+\Delta t}, \mu'^{\left(s_{i+\Delta t};\theta^{\mu'}\right)};\theta^{q'}\right) \quad (7)$$

Moreover, a soft update mechanism is performed for parameters update of target actor and target large networks. This process is formulated in Equation (8) as follows:

$$\begin{cases} \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)^{\theta^{\mu'}} \\ \theta^{q'} \leftarrow \tau\theta^{q} + (1-\tau)^{\theta^{q'}} \end{cases} \quad (8)$$

Where, $\theta^{\mu'}$ and $\theta^{q'}$ specifies a target actor as well as large network parameters individually. $\tau \in (0,1]$ specifies a number of soft update. A clipped Gaussian noise is extended for actions to enhance an arbitrary number of the control actions by using Equation (9) as follows:

$$\bar{a}(t) = a(s|\theta^{\mu}) + N \quad (9)$$

Here, N specifies a Gaussian procedure that designs a DDPG agent to most efficiently identify a constant action domain. Then, an approach begins to implement through irregular iteration and involves different phases such as exploration, understanding as well and convergence. Algorithm 1 represents the pseudocode of the DDPG approach for better reproducibility.

Algorithm 1: Pseudocode of DDPG approach

1. Initialize hyperparameters θ^{μ}, θ^{q} for actor and critic networks
2. Initialize hyperparameters for target actor as well as critic networks $\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{q'} \leftarrow \theta^{q}$
3. Initialize m-sized replay buffer D
4. For episode range from 1 to M do
5. Reset all environment for action identification
6. Perceive initial state
7. For $t = 1, \dots, T$; do
8. Produce an action based on present policy as well as sampled noise
9. Implement an action as well as acquire the reward as well as next state
10. Keep a transition $a(t), r(t), s(t), s(t+1)$ in a reply buffer D
11. If D is full then
12. Arbitrarily sample m-sized transitions
13. If episode is even then
14. Update θ^{μ} with Equation (5)
15. Update θ^{q} with Equation (6) and (7)
16. Update $\theta^{\mu'}$, $\theta^{q'}$ with Equation (18)
17. End if
18. End if
19. End for
20. End for

3.3.1 Partially Observable Markov Decision Process

This research concentrates on obtaining better preservation mechanisms for an individual-unit preservation issue. Various types of costs are incorporated into this decision-making problem,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

including the cost of acquiring task details during operation, the cost of performing preventive maintenance, and the cost of replacing or repairing a unit upon failure. The aim is to identify an optimal selection of actions that can be carried out at any point of a unit's operation. Mathematically, this research aims to address a consecutive decision-making issue that reduces the expected cost in a long execution.

Assume $V(\pi)$ represents an ideal anticipated overall discounted costs across an infinite horizon when a present acceptance state is π. The $V(\pi)$ is formulated in Equation (10) as follows:

$$V(\pi) = min\begin{cases} C_p + V(\pi^0) \quad for\ action = PM \\ C_o + \left(C_f + V(\pi^0)\right)\left(1 - R(\pi)\right) + \\ \Im\left(\sum_{y_i \in y} Pr(y_t|\pi)V(\pi, y_t)\right)R(\pi)\ for\ action = Operate \end{cases} \tag{10}$$

Here, action PM experiences a cost C_p and changes a system to new state. $\Im \in \{0,1)$ denotes the discount factor. The action termed "operate" encompasses multiple cost-related components, including the operational cost C_o, the potential cost arising from unforeseen failures, and the discounted expected cost associated with the next time step, assuming the unit remains functional until the following observation. This expected future cost is calculated as a weighted average, based on the probability of choosing to continue operating the unit after receiving an observation y_t.

3.4 Research Method Protocol

The complete research protocol implemented in this work involves following sequential stages:

Step1: VFC Environment Creation. A CloudSim-based VFC environment was designed which involving task vehicles, service vehicles, RSUs, and fog servers. Vehicle mobility, communication range, task size, and computational capacity were initialized based on parameters listed in Table 1.

Step 2: Task Generation and Classification. Computational tasks were produced through various data sizes, CPU requirements, priorities, and deadline constraints. Every task was classified as hard real-time, near real-time, soft real-time, or delay-tolerant.

Step 3: State and Action Modelling. The task offloading procedure was demonstrated as a POMDP. The state vector involved are vehicle position, available computing resources, task queue status, communication delay, and task priority. The action space comprised of local execution, RSU offloading, and service-vehicle offloading.

Step 4: Training of the POMDP-DDPG Agent. An actor and critic networks were trained through experience replay and target network updates. At every episode, the agent observed the environment, chosen an action, received a reward according to delay and energy consumption, and updated the policy.

Step 5: Performance Evaluation. The trained model was is estimated over RL, FL, DRL, and DDPG methods by mean energy consumption, latency ratio, completed tasks, packet loss, waiting time, service delay, and success rate.

Step 6: Comparative Analysis. Eventually, the results were compared to existing approaches for determining an effectiveness, robustness, and scalability of the proposed POMDP-DDPG approach.

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section deliberates an extensive simulation that is employed based on [21] benchmark dataset for various inconsistent scheduling parameters to evaluate an introduced. The simulation parameters are initially outlined, after that, a merging analysis of the introduced approach is taken out. Then, a comparative effectiveness estimation was attained for measured scheduling parameters over previous methods. The experiments are implemented by using the CloudSim simulator.

Table 1: Simulation parameters

Parameters	Values
Noise power	$1.5 \times 10^{-8}W$
Transmission power	$2 \pm 0.1W$
Learning Rate	0.01
Channel Gain	$144 \pm 1.44dB$
Computational complexity of Vehicle	0.3 $\pm$ 0.03 $Gigacycle/s$
Computational capacity of MFC server	5 Gigacycles
Communication capacity of MFC server	40 Mhz

There are various MFC servers in the proposed framework, and every MFC server is associated with an individual RSU, which is utilized in vehicular communication systems. Furthermore, each server in the system is equipped with multiple cores—independent processing units capable of executing tasks concurrently. Task lengths vary between 0 and 15,000 Million Instructions (MIs), with data sizes selected from the set {30, 35, 40, 45, 50, 60} MB. The computational resource demand for each task is randomly assigned from a set of values: {0.6, 0.8, 1.0, 1.2, 1.4} Gigacycles per second. Furthermore, each task is divided randomly into 6 to 10 subtasks for parallel or distributed execution. Transmission power is influenced by the system's underlying hardware. A transmission power of IoT devices depends on several hardware components, including the transceiver, power amplifier, processor, and power supply units. The existing methods such as RL, FL, DRL and DDPG, are estimated and compared with the proposed POMDP-DDPG approach. Table 1 demonstrates the simulation parameters utilized in this research.

4.1 Performance Analysis

Table 2 illustrates a performance estimation of mean energy consumption as well as maximum computational latency ratio based on the number of rounds. The POMDP-DDPG shows a consistent improvement in both energy consumption and latency compared to other baseline methods, validating its optimization advantage in cumulative execution rounds.

Table 2: Performance estimation of mean energy consumption and maximum computational latency ratio based on the number of rounds

Metrices	Methods	Number of rounds				
		100	200	300	400	500
Mean energy consumption (J)	RL	20	21	23	26	27
	FL	18	19	21	23	25
	DRL	15	16	17	18	19
	DDPG	13	14	14.5	15.5	16
	POMDP-DDPG	10	10.5	11	11.5	12
Maximum computational latency ratio	RL	0.88	0.89	0.91	0.92	0.94
	FL	0.84	0.85	0.87	0.88	0.90
	DRL	0.77	0.78	0.80	0.82	0.84
	DDPG	0.71	0.73	0.75	0.76	0.78
	POMDP-DDPG	0.65	0.66	0.67	0.68	0.70

Table 3: Performance estimation of proposed method based on number of vehicles

Metrices	Methods	Number of Vehicles				
		10	20	30	40	50
Completed Tasks (%)	RL	70	72	73	75	76
	FL	74	75	76	77	78
	DRL	80	82	84	85	87
	DDPG	83	85	87	88	89
	POMDP-DDPG	87	89	91	92	94
Packet Loss	RL	0.28	0.26	0.24	0.22	0.21
	FL	0.25	0.23	0.21	0.20	0.19
	DRL	0.18	0.16	0.15	0.14	0.13
	DDPG	0.14	0.13	0.12	0.11	0.10
	POMDP-DDPG	0.11	0.10	0.09	0.08	0.07
Average waiting time (ms)	RL	105	108	110	113	115
	FL	101	103	105	108	110
	DRL	95	96	97	99	100
	DDPG	90	91	92	93	94
	POMDP-DDPG	85	86	87	88	89
Average Delay (ms)	RL	185	190	195	198	200
	FL	178	182	185	188	190
	DRL	160	162	165	168	170

	DDPG	150	152	155	158	160
	POMDP-DDPG	140	142	145	148	150

Table 3 specifies a performance estimation of proposed method based count of vehicles. This table's results illustrate that POMDP-DDPG consistently performs better across increasing vehicle density by ensuring higher task completion, lower packet loss, and reduced delays due to smarter resource allocation.

Table 4 illustrates a performance estimation of proposed approach based on number of RSUs. A table reflects how increasing the number of RSUs improves service coverage and availability, which leads to lower delay and higher success rate for task execution. The POMDP-DDPG approach leverages this mobility and connectivity benefit more effectively through its advanced decision-making mechanism.

Table 4: Performance estimation of the proposed method based on the number of RSUs

Metrices	Methods	Number of RSUs				
		3	6	9	12	15
Service Delay (s)	L	1.25	1.22	1.18	1.15	1.12
	FL	1.18	1.15	1.12	1.09	1.07
	DRL	1.08	1.06	1.04	1.02	1.00
	DDPG	1.00	0.98	0.95	0.93	0.91
	POMDP-DDPG	0.90	0.88	0.85	0.83	0.80
Success Rate (%)	RL	75	76	77	78	79
	FL	78	79	80	81	82
	DRL	82	84	85	86	87
	DDPG	85	87	88	89	90
	POMDP-DDPG	88	89	91	92	94

4.2 Discussion

This research introduces a robust solution to the key limitations observed in existing vehicular fog computing task offloading models. While many previous studies focus on optimizing either energy consumption or latency in isolation, they often overlook the complex and dynamic nature of vehicular networks, particularly factors such as task dependencies, vehicle mobility, partial observability, and real-time execution constraints. The proposed POMDP-DDPG approach integrates reinforcement learning with POMDP to make adaptive decisions under uncertainty and incomplete state information. By incorporating actor-critic networks and experience replay, it achieves a stable and optimal policy learning process. Unlike static or heuristic methods, POMDP-DDPG learns from dynamic vehicular patterns, supports multi-metric optimization, and improves task allocation across RSUs and service vehicles. Simulation results demonstrate its superiority in minimizing energy consumption, latency, packet loss, and increasing task completion rates and service success. This substantiates the importance of intelligent, context-aware task offloading in future fog-enabled vehicular networks.

The major finding of this research is that modelling the task offloading problem as a partially observable sequential decision process importantly enhances an effectiveness of vehicular fog systems. As compared with existing RL-, FL-, DRL, and DDPG-based methods, the proposed approach generated most reliable and precise offloading decisions because it explicitly considers imperfect data and dynamic vehicular mobility. The proposed approach illustrated a 25% minimization in energy consumption and approximately 10–15% enhancement in latency-related measures when compared with the adjacent existing DDPG approach. Nevertheless, when compared with some advanced multi-agent and graph-based studies, the present work still has certain shortcomings. The present pipeline does not yet support collaborative learning among multiple RSUs, and it has not been estimated under highly heterogeneous urban traffic conditions with different intersections and differing communication advancements. Thus, although the proposed method outperforms existing baselines, its applicability to very large-scale smart-city scenarios remains an open research challenge.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.3 Difference from Prior Research and Significance of Proposed Method

The proposed POMDP-DDPG approach differs from existing studies in various significant perspectives. Existing approaches like RL, FL, DRL, and traditional DDPG significantly consider comprehensive state data and concentrate on one or two optimisation objectives. On the other hand, the proposed method captures the task offloading issue under partial observability, where the comprehensive model state may not always be accessible because of mobility, communication interruptions, or changing traffic density. As compared to existing method, the proposed pipeline simultaneously assumes task priority, delay constraints, energy consumption, packet loss, mobility of service vehicles, and dependency between the tasks. This allows an appeoach to make more practical offloading decisions in highly dynamic vehicular backgrounds. The POMDP formulation enhances the capability of the system to make consistent decisions even when the available data is incomplete, whereas the DDPG element continuously learns an optimal policy from the changing background.

Compared with traditional DDPG approach, the proposed POMDP-DDPG minimized mean energy consumption from 16 J to 12 J after 500 rounds, minimized a maximum computational latency ratio between 0.78 and 0.70, enhanced task completion from 89% to 94%, and minimized packet loss from 0.10 to 0.07. These enhancements proven that the proposed approach provides a more balanced and scalable solution than previous task offloading methods. However, the proposed approach also has certain limitations. The model needs a training phase through adequate interaction data, and the computational complexity of actor-critic learning process may exceed for large vehicular networks. Furthermore, the present framework considers a single-road scenario and homogeneous communication conditions. These aspects will be extended in future work.

5. CONCLUSION

Fog computing nodes are chosen across cloud computing servers for rapid dispensation of tasks. This research aimed to develop an intelligent task offloading pipeline for VFC which simultaneously reduces the latency, energy consumption, and packet loss whereas enhancing the task completion and service success under highly dynamic vehicular conditions. This research proposes the POMDP-DDPG approach for the task offloading in VFC, which purposes to attain a minimum latency as well as energy consumption. An approach captures the task offloading as a sequential decision-making problem in which incomplete system data, vehicle mobility, task priority, task dependency, and communication uncertainty are all considered. By the simulations, this research has validated the proposed method's mechanism in minimizing the latency and consumption over the different applications with the existing approaches. Compared with RL, FL, DRL, and conventional DDPG approaches, the proposed POMDP-DDPG achieved the minimum energy consumption of 12 J, minimum computational latency ratio of 0.70, the improved task completion rate of 94%, and reduced packet loss of 0.07.

The research question raised in the Introduction was whether an intelligent task offloading mechanism can significantly minimize latency and energy consumption in highly dynamic VFC environments where complete system data is not always available. The obtained findings clearly answer this question in the agreeing. Through integrating POMDP with DDPG, the proposed approach effectively maintained the uncertainty, vehicle mobility, and task heterogeneity, thus generating better offloading decisions than existing methods. Therefore, the research illustrtaes that uncertainty-aware reinforcement learning is a practical and efficient solution for next-generation vehicular fog systems.

Despite these advantages, some limitations remain. The simulation environment assumes only a single-road traffic scenario and considers constant communication among RSUs and vehicles. In practical large-scale urban networks, regular disconnections, heterogeneous vehicle densities, and different RSUs cause the model effectiveness. Furthermore, the training cost of the reinforcement learning model enhance significantly when the number of vehicles becomes very large. Therefore, future work will concentrate on improving the proposed pipeline to multi-road and multi-intersection scenarios, integrating heterogeneous communication technologies such as 5G and 6G, and developing lightweight distributed learning mechanisms for minimizing the computational overhead of training.

REFERENCES

[1] M.K. Farimani, S. Karimian-Aliabadi, R. Entezari-Maleki, B. Egger, and L. Sousa, "Deadline-aware task offloading in vehicular networks using deep reinforcement learning", *Expert Systems with Applications,* Vol. 249, 2024, p. 123622.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] J. Shi, J. Du, J. Wang, and J. Yuan, "Deep reinforcement learning-based V2V partial computation offloading in vehicular fog computing," *IEEE Wireless Communications and Networking Conference (WCNC),* 2021, pp. 1-6.

[3] A.M.A. Hamdi, F.K. Hussain, and O.K. Hussain, "Task offloading in vehicular fog computing: State-of-the-art and open issues", *Future Generation Computer Systems,* Vol. 133, 2022, pp. 201-12.

[4] F.G. Wakgra, B. Kar, S.B. Tadele, S.H. Shen, and A.U. Khan, "Multi-objective offloading optimization in MEC and vehicular-fog systems: A distributed-TD3 approach", *IEEE Transactions on Intelligent Transportation Systems,* Vol. 25, No. 11, 2024, pp. 16897-16909.

[5] K. Behravan, N. Farzaneh, M. Jahanshahi, and S. A.H. Seno, "A comprehensive survey on using fog computing in vehicular networks", *Vehicular Communications,* Vol. 42, 2023, p. 100604.

[6] M.A. Thanedar and S.K. Panda, "Energy and priority-aware scheduling algorithm for handling delay-sensitive tasks in fog-enabled vehicular networks", *The Journal of Supercomputing,* Vol. 80, No. 10, 2024, pp. 14346-14368.

[7] F. Dai, G. Liu, Q. Mo, W. Xu, and B. Huang, "Task offloading for vehicular edge computing with edge-cloud cooperation", *World Wide Web,* Vol. 25, No. 5, 2022, pp. 1999-2017.

[8] A. Hazra, P. Rana, M. Adhikari, and T. Amgoth, "Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges", *Computer Science Review,* Vol. 48, 2023, p. 100549.

[9] K. Moghaddasi, S. Rajabi, F. Soleimanian Gharehchopogh, and M. Hosseinzadeh, "An energy-efficient data offloading strategy for 5G-enabled vehicular edge computing networks using double deep Q-network", *Wireless Personal Communications,* Vol. 133, No. 3, 2024, pp. 2019-2064.

[10] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing", *IEEE Transactions on Intelligent Transportation Systems,* Vol. 24, No. 12, 2023, pp. 15513-15526.

[11] T.H. Binh, H. Vo, B.M. Nguyen, and H.T.T. Binh, "Reinforcement learning for optimizing delay-sensitive task offloading in vehicular edge–cloud computing", *IEEE Internet of Things Journal,* Vol. 11, No. 2, 2023, pp. 2058-2069.

[12] W. Mao, O.U. Akgul, B. Cho, Y. Xiao, and A. Ylä-Jääski, "On-demand vehicular fog computing for beyond 5G networks", *IEEE Transactions on Vehicular Technology,* Vol. 72, No. 12, 2023, pp. 15237-15253.

[13] J. Lin, S. Huang, H. Zhang, X. Yang, and P. Zhao, "A deep-reinforcement-learning-based computation offloading with mobile vehicles in vehicular edge computing", *IEEE Internet of Things Journal,* Vol. 10, No. 17, 2023, pp. 15501-15514.

[14] V. Jain and B. Kumar, "QoS-aware task offloading in fog environment using multi-agent deep reinforcement learning", *Journal of Network and Systems Management,* Vol. 31, No. 1, 2022, p. 7.

[15] A. Robles-Enciso and A. F. Skarmeta, "A multi-layer guided reinforcement learning-based tasks offloading in edge computing", *Computer Networks,* Vol. 220, 2023, p. 109476.

[16] Y. Cheng, A. Vijayaraj, K.S. Pokkuluri, T. Salehnia, A. Montazerolghaem, and R. Rateb, "Vehicular fog resource allocation approach for VANETs based on deep adaptive reinforcement learning combined with heuristic information", *IEEE Access,* Vol. 12, 2024, pp. 139056-139075.

[17] G. Liu, F. Dai, B. Huang, Z. Qiang, S. Wang, and L. Li, "A collaborative computation and dependency-aware task offloading method for vehicular edge computing: A reinforcement learning approach", *Journal of Cloud Computing,* Vol. 11, No. 1, 2022, p. 68.

[18] D.H. Abdulazeez and S.K. Askar, "Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing environment", *IEEE Access,* Vol. 11, 2023, pp. 12555-12586.

[19] K. Moghaddasi, S. Rajabi, and F.S. Gharehchopogh, "Multi-objective secure task offloading strategy for blockchain-enabled IoV-MEC systems: A double deep Q-network approach", *IEEE Access,* Vol. 12, 2024, pp. 3437-3463.

[20] L. Geng, H. Zhao, J. Wang, A. Kaushik, S. Yuan, and W. Feng, "Deep-reinforcement-learning-based distributed computation offloading in vehicular edge computing networks", *IEEE Internet of Things Journal,* Vol. 10, No. 14, 2023, pp. 12416-12433.

[21] K. Mishra, G.N. Rajareddy, U. Ghugar, G.S. Chhabra, and A.H. Gandomi, "A collaborative computation and offloading for compute-

intensive and latency-sensitive dependency-aware tasks in dew-enabled vehicular fog computing: A federated deep Q-learning approach", *IEEE Transactions on Network and Service Management,* Vol. 20, No. 4, 2023, pp. 4600-4614.

[22] V. Sethi and S. Pal, "FedDOVe: A federated deep Q-learning-based offloading for vehicular fog computing", *Future Generation Computer Systems,* Vol. 141, 2023, pp. 96-105.

[23] B. Jamil, H. Ijaz, M. Shojafar, and K. Munir, "IRATS: A DRL-based intelligent priority and deadline-aware online resource allocation and task scheduling algorithm in a vehicular fog network", *Ad Hoc Networks,* Vol. 141, 2023, p. 103090.

[24] Z. Wei, B. Li, R. Zhang, X. Cheng, and L. Yang, "Many-to-many task offloading in vehicular fog computing: A multi-agent deep reinforcement learning approach", *IEEE Transactions on Mobile Computing,* Vol. 23, No. 3, 2023, pp. 2107-2122.

[25] N. Chen, P. Zhang, N. Kumar, C.H. Hsu, L. Abualigah, and H. Zhu, "Spectral graph theory-based virtual network embedding for vehicular fog computing: A deep reinforcement learning architecture", *Knowledge-Based Systems,* Vol. 257, 2022, p. 109931.

[26] A.N. Alvi, M.A. Javed, M.H.A. Hasanat, M.B. Khan, A.K.J. Saudagar, M. Alkhathami, and U. Farooq, "Intelligent task offloading in fog computing based vehicular networks", *Applied Sciences,* Vol. 12, No. 9, 2022, p. 4521.

[27] M.R. Raju, S.K. Mothku, and M.K. Somesula, "MITS: Mobility-aware intelligent task scheduling in vehicular fog networks", *IEEE Transactions on Vehicular Technology,* Vol. 73, No. 3, 2023, pp. 3079-3093.

[28] S. Pang, L. Hou, H. Gui, X. He, T. Wang, and Y. Zhao, "Multi-mobile vehicles task offloading for vehicle-edge-cloud collaboration: A dependency-aware and deep reinforcement learning approach", *Computer Communications,* Vol. 213, 2024, pp. 359-371.

[29] X. Yin and M. Lei, "Jointly improving energy efficiency and smoothing power oscillations of integrated offshore wind and photovoltaic power: A deep reinforcement learning approach", *Protection and Control of Modern Power Systems,* Vol. 8, No. 2, 2023, pp. 1-11.