Computational offloading in vehicular edge computing using multiple agent-based deterministic policy gradient algorithm and generative adversarial networks

Abstract

The development of the Intelligent Connected Vehicles (ICV) and Internet of Vehicles (IoV) as an evolving technology has changed the Vehicular Edge Computing environment. Computational offloading is the primary challenge in Edge Computing. Effective computation offloading algorithms improve computing performance, network resource utilization and energy consumption. Traditional computation offloading algorithms are based on local states and the decisions are not focused on vehicular edge network utilization. Although numerous offloading algorithms are proposed to achieve computing performance, the mobility and offloading failure are rarely considered for optimization and it remains challenging. To address the challenge, this article presents Computational Offloading in Vehicular Edge Computing using Multiple Agents based Deterministic Policy Gradient Algorithm and Generative Adversarial Networks. The optimization problem as a minimization problem is defined to reduce system overhead. The processing time delay of offloading and energy consumption are introduced in defining the problem. The proposed system increases the number of offloading executions with a minimum number of edge servers based on Deterministic Policy Gradient. To provide interaction between vehicle and edge servers, Multiple Agents, Deep Reinforcement Learning and Markov decision process with five attributes are used. The Generative Adversarial Networks in the actor-critic network and Deterministic Policy Gradients increases the learning rate and efficiency of the system. Simulation and its results show the system gives minimal system overhead while the number of iterations increases and is subject to processing time delay and energy utilization.

Keywords: Multiple Agents, Deep Reinforcement Learning, Internet of Vehicles, Intelligent Connected Vehicles, Computation offloading, Vehicular Edge Computing, Processing time delay, Energy utilization, Deterministic Policy Gradient, Generative Adversarial Networks and Network resource utilization

Introduction

The evolution of mobile computing technologies like fifth generation minimize the re-transmission speed. With rapid development of integrated circuits, the chip sizes are reduced to micro level and the computational power and storage size of the circuits are increased to macro level. Enormous usage of smart devices moves human society to the world of Internet for Everything. In a multi-disciplinary system, the Internet of Things is used for real time data processing. Smart mobile devices and Smart edge devices are popularized with different applications such as object detection, unified payment, navigation and Augmented Reality based applications [2]. The traditional computing environment used a cloud computing model which all uses centralized or distributed servers. Auto scaled cloud computing services cannot effectively handle computational tasks shared by smart edge nodes.

To address the challenge, the Edge computing paradigm moves the computational power closer to edge nodes. A new computing paradigm minimizes the response time of computation, network bandwidth to access the cloud server and maximizes the Service given to the edge users. Internet Connected Vehicle (ICV) and Edge computing (EC) are merged in Vehicular Edge Computing (VEC) in which a computational unit is processed on the Vehicles or offloading to VEC Edge Server in the network .

Computation offloading refers to the method of transferring computation intensional tasks to a system located in the edge layer or cloud layer of the Vehicular Edge network.The existing computation offloading methods offload the task to cloud computing platforms which perform the computation with higher end systems and it effectively reduces the system overhead [5]. Due to the longest data transmission between Vehicle and Cloud server gives maximum processing time delay and energy utilization. The task processing time and energy utilization of a vehicle destroy the utilization of intelligent applications in Edge Computing [6].

To address the challenge and minimize the system overhead, a Computation Offloading algorithm based on Multiple Agents based Deterministic Policy Gradient Algorithm and Generative

Adversarial Networks is proposed in this paper. The primary objectives of the system are listed below.

- The problem of computation offloading is modeled as a discrete stochastic control process (i.e.) Markov decision process with five attributes.
- Multiple Agent based Deterministic Policy Gradient Algorithm and Generative Adversarial Networks are used to minimize the system overhead subject to processing time delay and energy consumption.
- Experiments were conducted in a simulated environment and the results compared with existing algorithms, which indicates the efficiency of the proposed methodology with minimum system overhead.

Related Work

Pu et al. [7], simulated a multi-task offloading in which a task centric scheduling system is implemented to minimize the energy utilization. Different numbers of vehicles are considered numerous vehicles in the offloading algorithm. The simulation and its experimental results illustrate performance of the scheduling system is improved due to the inclusion of a task centric scheduling algorithm.

Guo et al. [8], designed resource scheduling and dynamic offloading algorithms to optimize the energy utilization and task processing time. Based on task related requirements and task execution time related characteristics, optimization linear programming problem as minimization defined to improve energy saving aspects. The objective function includes offloading selection, frequency related parameter and power allocation. The framework achieves energy utilization efficiently and the efficiency of the system analyzed in a heterogeneous environment.

Zhai et al. [9], discussed the computing device utilized energy is limited and the energy utilization depends on the working time of the system. The heuristic algorithm is used to experiment the energy aware offloading process. Software defined networks and fog computing architecture are considered for improving the Internet of Vehicle network dynamics. The objective function is defined with an execution cost model which includes battery power as a dynamic weight factor. The dependency between the applications are analyzed in the experiments. The result indicates the offloading algorithm executes numerous applications with respect to available power in the battery.

Wang et al. [10], solved the offloading decision problem using dynamic reinforcement learning algorithm. The parameters like network traffic and computational tasks are considered. At the same time the dependency between tasks is not accounted for. In the vehicular network, energy utilization and delay in the service time are analyzed in the simulated experimental setup which shows the vehicular edge network. The existing algorithms are implemented to prove the efficiency of an algorithm. The enumerated results indicate the dynamic reinforcement learning algorithm gives better results.

Ren et al. [11], addressed an offloading strategy using DDPG [3] to minimize the service based on multiple constraints. The mobility related parameters in the transport, heterogeneous systems were considered for implementation. The simulation results indicate the DDPG system improves the performance of the vehicular edge network and is stable compared to other algorithms.

Haitao et al. [12], implemented a DQN based task offloading algorithm. The MEC server's computational power is infinite in the architecture.The tasks are assigned with priority based on an analytic hierarchy process which provides various weight factors to the mode. The objective of the minimization function is to reduce the task processing rate. To analyze the task processing aspects of a DQN algorithm, Q Learning is implemented. The result indicates the task of offloading efficiency is improved significantly.

Liu et al. [13], studied Vehicular Edge Computing architecture and proposed an effective computation offloading strategy. The optimization problem given in the form of maximization. The traffic of the network is randomly chosen for maximizing the network utilization. The Semi markov decision process model is integrated with the Q Learning algorithm. To maximize the network utilization in computational offloading, deep reinforcement learning models are used.

Du et al. [14], integrated Deep Learning and Long Short Term Memory (LSTM) for achieving better performance and solving road planning problems. The resource allocation problem is evaluated using convolutional networks and memory. Multiple agents based markov decision process models are formulated to express the efficiency [15].

Ning et al. [16], addressed a multi decision approach to optimize the total task delay. The offload the task optimization problem is defined in the form of nonlinear programming problem for minimizing the processing delay subject to energy constraints. The proposed strategy increases the performance of the edge architecture and Quality of Services (QoS).

Dong et al. [17], discussed the deep learning used offloading strategy for edge computing. The model considers resource and task allocation processes in edge computing. The optimization problem is defined in terms of maximization problem for Quality of Experience subject to resource and task allocation constraints. An intelligent offloading strategy implemented using the DQN algorithm. The experimental result shows that Quality of Experience is maximized by using the DQN algorithm.

Flores et al. [18], addressed the framework for mobile and cloud computing models. The mobile application task is divided into minimal size sub tasks in the framework. The tasks are analyzed and the most intensive tasks are offloaded to the instances running in the cloud to improve the overall system overhead. The framework is used for data binding and offloading processes.

Zhang et al. [19], enumerated a proximal policy optimization based on deep reinforcement and markov decision process model to solve optimization problems. The convolution neural networks integrated with network architecture to utilize the policy in vehicular edge computing architecture. The training efficiency improved by state and rewards. The simulation results are compared to existing algorithms to indicate the advantage of a proximal policy optimization algorithm.

Xu et al. [20], implemented a method called adaptive computational offloading. The optimization function is defined in the form of resource utilization and time delay. The 5G network between Vehicle and Road Side Units are used to improve the efficiency of the transmission. Evolutionary algorithms are incorporated to generate the solutions. The experimental results indicate that the system performance is maximized.

Yao et al. [21], constructed an offloading problem with the help of Markov Decision Process. A twin delayed algorithm is used for computational offloading. The system integrates Internet of Vehicles with deep reinforcement learning models. Energy consumption is considered to define the optimization problem. The simulated environment evaluates the viability of the system.

Falahatraftar et al. [22], studied dynamic heterogeneous vehicular networks with 5G networks. Network functions based architecture and software defined networks are proposed for vehicular networks. The Condition based Generative adversarial network is used to generate the vehicular network scenario. The experimental result shows that the proposed network trained to generate the data which resembles real data.

Zhao et al. [23], implemented vehicular networks using Generative adversarial networks. The model is adapted to identify the next location of the vehicle. The coordinate specific transformation models are incorporated in the experiments. The results indicate the network gives better efficiency in terms of absolute error and accuracy.

Qianqian et al. [24], proposed a framework for an unmanned aerial vehicle using Generative adversarial network. The neural network is used to train the local channel model that enables learning the data in a distributed approach. The usage of Generative adversarial network generates more samples in each episode. The result indicates the architecture gives best learning accuracy. Generative adversarial networks is an efficient algorithm for generating the data from learned data [25].

System Model

In Vehicular Edge computing network architecture, each VEC Edge Server serves the vehicle in a cell via Road Station Unit (RSU). The proposed system focuses on effective computation offloading between the Vehicles and Vehicle to RSU in a cell.

Let V be the set of vehicles in VEC architecture. The vehicle in the bottom layer generates a task in the form of object detection and an offloading request is submitted to VEC Edge Server via RSU to perform computation offloading. The RSU network coverage is represented in the form of hexagon and its diameter is 2*L where L is the side of the cell.

In the VEC architecture, The computation offloading index Oi represents the decision taken by the offloading process given in equation (1)

$Oi\ = \{\ 0\ No\ computation\ offloading, 1\ offloading\ to\ VEC\ Edge\ Server\ via\ RSU$ -- (1)

First case, No computation offloading refers to all the tasks being computed on Vehicle. In the Second case, The successful computation of computational tasks by offloading to VEC Edge Servers using Road Side Units.

If a computational offloading index is 0, then all the tasks are completely computed in the Vehicle Control Unit (VCU). The delay and Energy consumption calculated as follows

D1 = (Tij * C) / F -- (2)

Where

D1 is Delay in No Computation offloading

Tij is Task generated by ith Vehicle jth time slot

C is CPU clock cycles required for computation

F is Frequency of Vehicle Control Unit

$EC1 = ((Tij * C) / F) * P1$ --(3)

Where

EC1 is Energy Consumption in No Computation offloading

P1 is Power consumption by the Vehicle in in No Computation offloading

If the computation offloading index is 1, then all the tasks are completely executed in the VEC Edge Server. The Energy consumption and processing delay are calculated as follows

$D2 = Oi\,(Tij * C) / F_{VEC\ Edge\ Server}$ -- (4)

Where

D2 is Delay in Computation offloading via RSU

Tij is Task generated by ith Vehicle jth time slot

C is CPU clock cycles required for computation in VEC Edge Server

F is Frequency of VEC Edge Server

$EC2 = Oi((Tij * C) / F_{VEC\ Edge\ Server}) * P2$ --(5)

Where

EC2 is Energy Consumption in Computation offloading via RSU

P2 is Power consumption by the VEC Edge Server in Computation offloading

The overall delay and Energy consumption of VEC calculated as follows

$D_{OVERALL} = \text{Maximum}\,(D1, D2)$ -- (6)

$EC_{OVERALL} = EC1 + EC2$ -- (7)

Problem Definition

For the computation offloading process, It is desirable to optimize the system overhead through Multi Agent Deterministic Policy Gradient Algorithm and Generative Adversarial Networks given in the following equation (8).

Minimize $Z = \sum_{\square=1}^{\square} \sum_{\square=1}^{\square}$ Aij.[Dw*DOVERALL + ECw*ECOVERALL] -- (8)

subject to

Dw + ECw = 1,

DOVERALL <= DMAX,

For all Dw and ECw ϵ [0,1],

For all Aij ϵ {0,1},

Where Z is system overhead,

Dw and ECw is delay weight and Energy Consumption weight

DMAX is Maximum delay of current task

Aij is the flag refers to jth task offload or not during ith time slot

k is overall time

m is total number of task

Computation offloading using Multiple Agents based Deterministic Policy Gradient Algorithm and Generative Adversarial Networks

Deep Reinforcement Learning has multiple agents, state, environment, reward and action. It is mathematically modeled as the Markov Decision Process [1]. VEC architecture and computation offloading model is implemented. In the proposed system, during the learning phase, multiple agents observe the reward and the result is transferred to the next environment. During the time period ‘i’, vehicles are present in the initial state Si and the deterministic policy implements an Action Ai. In the edge environment, modification is done based on the action and enters into new state Si+1. At the same time, The reward Ri returned to the Edge computing environment. The bottom tier vehicle turns up with a revised policy from state Si+1 and continues with an Edge

Environment. The VEC architecture generates a huge amount of data using a generative adversarial network. The policy adjusted with the help of data generated by the VEC environment. At the end, the VEC architecture learns the optimized policy that maximizes the return of the task offloaded.

Based on the Markov decision process model, the optimization problem given in equation (8) is solved using multiple agents based deep deterministic policy gradients and generative adversarial networks. To generate a huge amount of data that resembles the training data, a generative adversarial network is introduced with the actor-critic network. The discriminator identifies the positive data from generative phase data. The neural network learning rate is improved by generative adversarial networks. The system uses continuous action and achieves better optimality.

The global state is considered in centralized DQN[4]. Each vehicles offloading decision needs agent observation. In the centralized decision process, an agent fails, the entire system affected.To overcome the issue, multiple agents are considered and each vehicle assigned with a separate agent and offloading decision is made by separate agents specific to the vehicles.

The relationship between multiple agents is described by the reward function. Each agent integrated with a separate reward function. In the actor-critic network, the actor defines the policies and generates actions based on generative adversarial networks. The critical phase evaluates the policy based on rewards obtained from the VEC environment.

The optimal policy identification is done in the Edge layer. Various input parameters of the algorithm includes VEC environment parameters like Neural Network, Number of hidden layers, Learning rate, Episode, VEC maximum calculation capability, Number of Vehicles, Processing time delay factor, Energy consumption factor, Frequency of Vehicle, Frequency of VEC Edge Server, Task Size and CPU clock cycles required for computation.

Algorithm

//Computation offloading using Multiple Agents based Deterministic Policy Gradient Algorithm and Generative Adversarial Networks

```
//Input: VEC environment parameters, Overall time k and Initiala state Si
//Output: Computation offloading decision with minimized system overhead
Initialize VEC environment and generate task requests from vehicles
if ( tasks == dividable)
        if ( vehicle range <= VEC Edge Server range)
                For each episode 1 to M (Maximum number of Episodes) do
                        The state and actions are initialized
                        Reinitialize VEC environment parameters and observation state
                        For each time period 1 to k do
                        Finalize the state Si and implement the action part
                        Find the reward from multiple agents and identify new state Si+1
                Process via Generative Adversarial Networks
                Calculate Q-Value of the network
                Update weight of policy network
                Update target Q-Value of the network
                        End For
                End For
Calculate system overhead of Computation offloading via RSU and Computation offloading between vehicles
Compare the decision  and finalize optimal computation offloading
        End If
End If
```

Table 1 VEC Environment simulation parameters

S. No	Parameter	Values
1.	Neural network	Generative Adversarial Networks
2.	Number of layers	3

3.	Number of neutrons	50
4.	Learning rate	6e-6
5.	Episode	1000
6.	VEC maximum calculation capability	10 GigaBits
7.	Number of Vehicles	15
8.	Processing time delay factor	0.5
9.	Energy consumption factor	0.5
10.	Vehicle's Frequency	0.7 to 1.25 GHz
11.	VEC Edge Server's Frequency	2.4 GHz
12.	CPU clock cycles required for computation	900
13.	Task Size in MegaBytes	5, 10, 15, 20, 25 and 30

Experimental Result and Analysis

The experiment was conducted on a 12th Gen Intel Core i7 Processor 3.90 GHz, 16 GB DDR3 RAM, 1.5 TB SSD and 64-bit Windows 8.1 operating system. A VEC Environment with two VEC Edge servers and 15 vehicles simulated using Python 3.10. The proposed algorithm tested on Tensorflow 2. Table 1 lists the simulation parameters of VEC Environment. Setting up the values for the simulation parameters is essential. The parameter setting will change the performance of the algorithm.

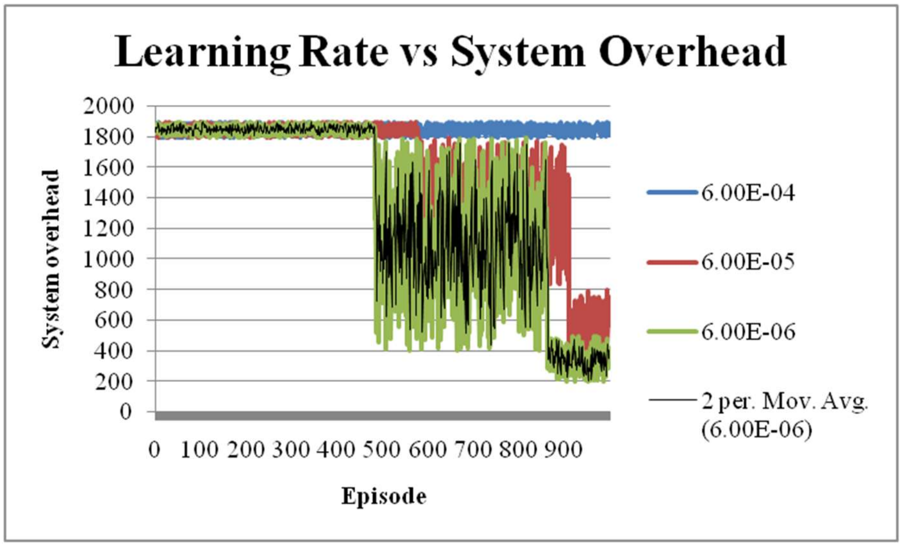

Figure 1 Learning Rate vs System Overhead

The system overhead reached its minimum while the neural network's learning rate is 6e-6. A local optimum is attained between 300 and 400 episodes. The batch size is defined as 128. In the Deep Neural Network, one hidden layer consists of 50 neutrons and Rectified Linear Units. To analyze the performance of the proposed Multiple Agents based Deterministic Policy Gradient algorithm and Generative Adversarial Networks, Deep Deterministic Policy Gradient, Distributed offloading scheme using DQN, Simple DQN and LSTM based DQN Algorithms were implemented.

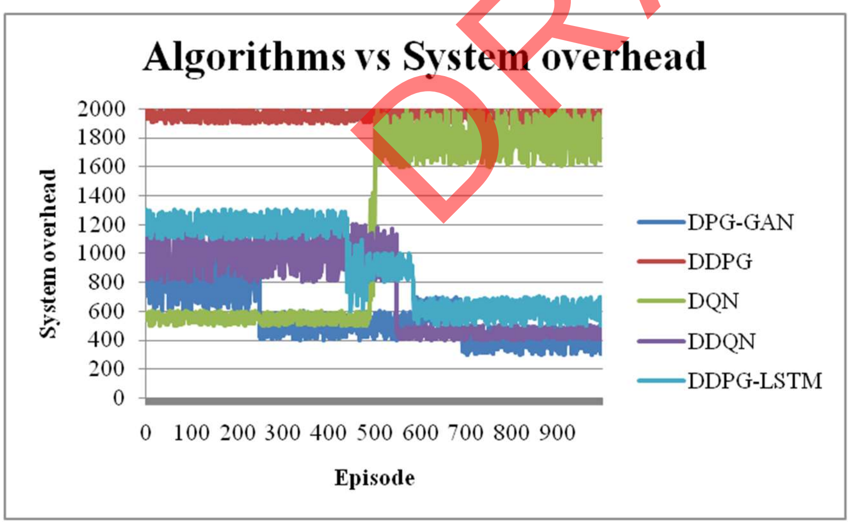

Figure 2 Algorithms vs System overhead

The figure 2 indicates the proposed algorithm gives minimum system overhead when the number of episodes increases. DDPG algorithm gives maximum system overhead in all the iterations. The Distributed DQN in minimal system overhead from 550 episodes. DDPG with LSTM gives an optimal system from 400 episodes, but it gives higher system overhead compared to the proposed system. In the proposed system, adding a generative adversarial network can improve the performance.

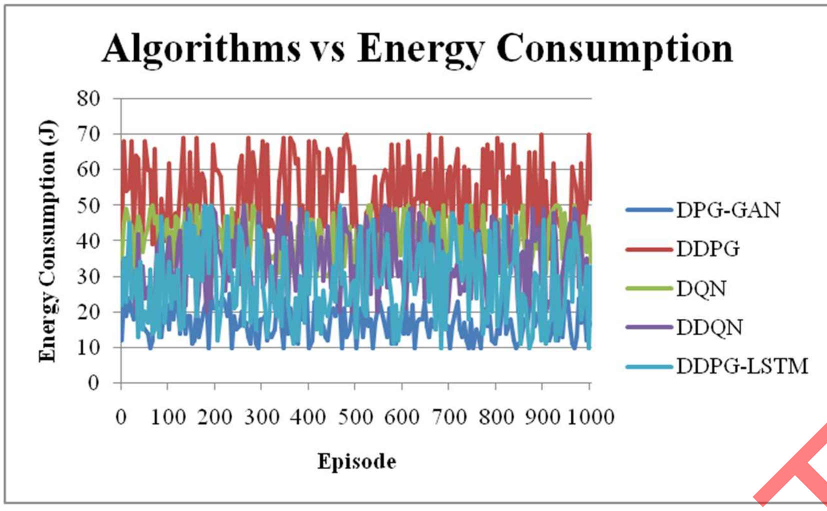

Figure 3 Algorithms vs Energy Consumption

The figure 3 indicates the proposed algorithm gives minimum energy consumption when the number of episodes increases. DDPG algorithm gives higher energy consumption in all the iterations. The Distributed DQN in minimal energy consumption compared to DQN and DDPG. DDPG with LSTM gives energy consumption up to 50J which is higher energy consumption compared to the proposed system.

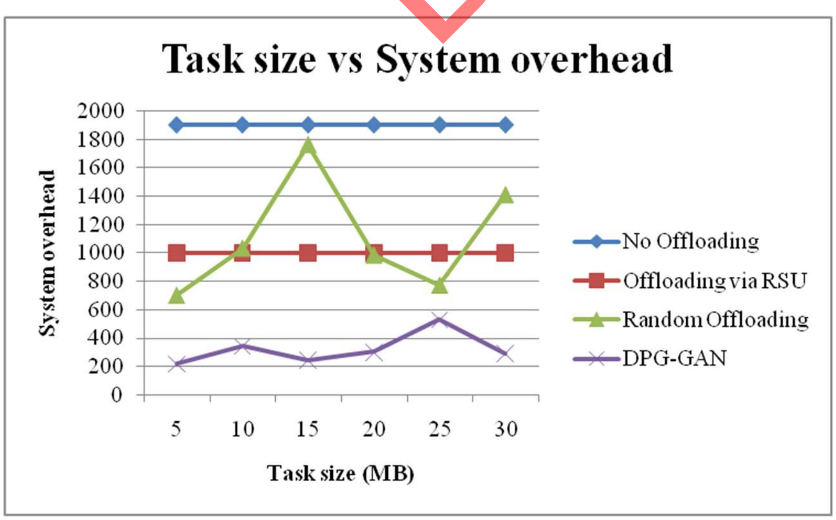

Figure 4 Task size vs System overhead

The proposed system compared with No Offloading, Offloading via RSU and Random Offloading conditions. Random offloading condition is done by randomly selecting No offloading and offloading via RSU cases. No offloading i.e computation in data generation layer gives highest system overhead. Random offloading gives 700 to 1800 system overhead due to random selection. Offloading via RSU gives constant system overhead which is 1000. The proposed system gives optimal system overhead in different sizes which lies between 200 and 450. Therefore, the Multi Agent Deterministic Policy Gradient Algorithm and Generative Adversarial Networks gives minimal system overhead and energy consumption in terms of number of iteration increases and the size of the task increases.

Conclusion

In Vehicular Edge computing architecture, the service capability of Internet Connected Vehicles is improved by better offloading algorithms. The linear programming problem is defined to minimize the system overhead. A minimization function considered processing time delay and energy utilization to achieve this target. The existing deep reinforcement algorithms such as DQN, DDQN and DDPG with LSTM are used to resolve discrete models. However, Computation offloading using Multiple Agent based Deterministic Policy Gradient Algorithm and Generative Adversarial Networks focus on continuous models. The optimal offloading decision is taken by VEC Edge Server with minimal processing time delay and energy utilization. The generative adversarial networks included into the actor-critic network generates huge amounts of data that resembles the training data. The discriminator generative adversarial network identifies the positive data from the generative phase. Multiple Agents and Generative Adversarial Networks are introduced to increase the efficiency. The simulated environment and experimental result shows the system gives minimal system overhead while the number of iterations increases and is subject to processing time delay and energy consumption. Collaborative offloading decisions and hybrid offloading strategies with minimum system overhead will be implemented as future work in Vehicular Edge Computing architecture by considering various task related characteristics and network related parameters.

References

1. Wang, Kun & Wang, Xiaofeng & Liu, Xuan & Jolfaei, Alireza. (2020). Task Offloading Strategy Based on Reinforcement Learning Computing in Edge Computing Architecture of Internet of Vehicles. IEEE Access. 8. 173779-173789. 10.1109/ACCESS.2020.3023939.
2. Kevin Ashton. (2009). That 'Internet of things' thing. RFiD J. 22(7), 97-101
3. Y. Ren, A. Guo, C. Song and Y. Xing, (2021), Dynamic Resource Allocation Scheme and Deep Deterministic Policy Gradient-Based Mobile Edge Computing Slices System. IEEE Access, vol. 9, pp. 86062-86073, 2021, doi: 10.1109/ACCESS.2021.3088450.
4. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236.
5. Lin, Bing & Lin, Kai & Lin, Changhang & Lu, Yu & Huang, Ziqing & Chen, Xinwei. (2021). Computation offloading strategy based on deep reinforcement learning for connected and autonomous vehicle in vehicular edge computing. Journal of Cloud Computing. 10. 10.1186/s13677-021-00246-6.
6. Liu, Lei & Pei, Qingqi & Maharjan, Sabita & Zhang, Yan. (2021). Vehicular Edge Computing and Networking: A Survey. Mobile Networks and Applications. 26. 10.1007/s11036-020-01624-1.
7. Pu, L., Chen, X., Mao, G., Xie, Q., & Xu, J. (2019). Chimera: An Energy-Efficient and Deadline-Aware Hybrid Edge Computing Framework for Vehicular Crowdsensing Applications. IEEE Internet of Things Journal, 6, 84-99.
8. Guo, S., Xiao, B., Yang, Y., & Yang, Y. (2016). Energy efficient dynamic offloading and resource scheduling in mobile cloud computing. IEEE International Conference on Computer Communications, 1-9.
9. Zhai, Y., Sun, W., Wu, J., Zhu, L., Shen, J., Du, X., & Guizani, M. (2020). An Energy Aware Offloading Scheme for Interdependent Applications in Software-Defined IoV With Fog Computing Architecture. IEEE Transactions on Intelligent Transportation Systems, 22, 3813-3823.

10. Wang, Y., Wang, K., Huang, H., Miyazaki, T., & Guo, S. (2019). Traffic and Computation Co-Offloading With Reinforcement Learning in Fog Computing for Industrial Applications. IEEE Transactions on Industrial Informatics, 15, 976-986.
11. Ren, Yinlin & Yu, Xiuming & Chen, Xingyu & Guo, Shaoyong & Xuesong, Qiu. (2020). Vehicular Network Edge Intelligent Management:A Deep-Deterministic Policy Gradient Approach for Service-Offloading Decision. 905-910. 10.1109/IWCMC48107.2020.9148507.
12. Haitao ZHAO,Tangwei ZHANG,Yue CHEN,Houlin ZHAO,Hongbo ZHU. (2020). Task distribution offloading algorithm of vehicle edge network based on DQN[J]. Journal on Communications, 41(10), 172-178. 10.11959/j.issn.1000-436x.2020160.
13. Liu, Yi & Yu, Huimin & Xie, Shengli & Zhang, Yan. (2019). Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks. IEEE Transactions on Vehicular Technology. PP. 1-1. 10.1109/TVT.2019.2935450.
14. Du, Yiquan & Zhang, Xiuguo & Cao, Zhiying & Wang, Shaobo & Liang, Jiacheng & Zhang, Fengge & Tang, Jiawei. (2021). An Optimized PathPlanning Method for Coastal Ships based on Improved DDPG and DP. Journal of Advanced Transportation. 2021. 10.1155/2021/7765130.
15. Chen, Juan & Xing, Huanlai & Xiao, Zhiwen & Xu, Lexi & Tao, Tao. (2021). A DRL Agent for Jointly Optimizing Computation Offloading and Resource Allocation in MEC. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2021.3081694.
16. Ning, Z., Zhang, K., Wang, X., Guo, L., Hu, X., Huang, J., Hu, B., & Kwok, R.Y. (2021). Intelligent Edge Computing in Internet of Vehicles: A Joint Computation Offloading and Caching Solution. IEEE Transactions on Intelligent Transportation Systems, 22, 2212-2225.
17. Dong, Peiran & Wang, Xiaojie & Rodrigues, Joel. (2019). Deep Reinforcement Learning for Vehicular Edge Computing: An Intelligent Offloading System. ACM Transactions on Intelligent Systems and Technology. 10. 10.1145/3317572.
18. Flores, Huber & Srirama, Satish & Buyya, Rajkumar. (2014). Computational Offloading or Data Binding? Bridging the Cloud Infrastructure to the Proximity of the Mobile User.

Proceedings - 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2014. 10-18. 10.1109/MobileCloud.2014.15.

19. Zhan, Wenhan & Luo, Chunbo & Wang, Jin & Wang, Chao & Min, Geyong & Duan, Hancong & Zhu, Qingxin. (2020). Deep Reinforcement Learning-Based Offloading Scheduling for Vehicular Edge Computing. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2020.2978830.
20. Xu, Xiaolong & Zhang, Xing & Liu, Xihua & Jiang, Jielin & Qi, Lianyong & Bhuiyan, Md. (2020). Adaptive Computation Offloading With Edge for 5G-Envisioned Internet of Connected Vehicles. IEEE Transactions on Intelligent Transportation Systems. PP. 1-10. 10.1109/TITS.2020.2982186.
21. Yao, Liang & Xu, Xiaolong & Bilal, Muhammad & Wang, Huihui. (2022). Dynamic Edge Computation Offloading for Internet of Vehicles With Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems. 10.1109/TITS.2022.3178759.
22. Falahatraftar, Farnoush & Pierre, Samuel & Chamberland, Steven. (2021). A Conditional Generative Adversarial Network Based Approach for Network Slicing in Heterogeneous Vehicular Networks. Telecom. 2. 141-154. 10.3390/telecom2010009.
23. Zhao, Liang & Liu, Yuefei & Al-Dubai, Ahmed & Zomaya, Albert & Min, Geyong & Hawbani, Ammar. (2020). A Novel Generation-Adversarial-Network-Based Vehicle Trajectory Prediction Method for Intelligent Vehicular Networks. IEEE Internet of Things Journal. 10.1109/JIOT.2020.3021141.
24. Zhang, Qianqian & Ferdowsi, Aidin & Saad, Walid. (2021). Distributed Generative Adversarial Networks for mmWave Channel Modeling in Wireless UAV Networks. 1-6. 10.1109/ICC42927.2021.9501056.
25. Alhussain, A., & Lin, M. (2022). Hardware-Efficient Deconvolution-Based GAN for Edge Computing. 2022 56th Annual Conference on Information Sciences and Systems (CISS), 172-176.