Article

Leveraging Time-Critical Computation and AI Techniques for Task Offloading in IoV Network Applications

Peifeng Liang * , Wenhe Chen , Honghui Fan * , Hongjin Zhu

Posted Date: 22 July 2024

doi: 10.20944/preprints202407.1624.v1

Keywords: time-critical; fog computing; deep learning; internet of vehicles; task offloading

Article

Leveraging Time-Critical Computation and AI Techniques for Task Offloading in IoV Network Applications

Peifeng Liang[1]*, Wenhe Chen[1] and Honghui Fan[1]* and Hongjin Zhu [1]

[1] School of Computer Engineering, Jiangsu University of Technology,Changzhou, Jiangsu, 213001, China
* Correspondence: lgod9@jsut.edu.cn; fanhonghui@jsut.edu.cn

Abstract: The study of Internet of Vehicle (IoV) based on Fog Computing (FC) and Artificial Intelligent (AI) has attracted more and more attention. However, there are still many problems require to investigate such as time-critical, scalability, load-balancing, energy consumption, and so on. Focusing on these problems, we proposed an AI-based Vehicle-to-Everything (V2X) model for tasks and resource offloading model for IoVs network, which ensures reliable low-latency communication efficient tasks offloading in IoV network by using Software Defined Vehicular based FC (SDV-F) architecture. To fit to time-critical data transmission task distribution, the proposed model reduces unnecessary task allocation at the fog computing layer by proposing an AI-based task-allocation algorithm in IoV layer to implement task allocation of each vehicle. By applying AI technologies of Reinforcement Learning (RL), Markov decision process, and Deep Learning (DL), the proposed model intelligently makes decision on maximizing resource utilization at the fog layer, and minimizing the average end-to-end delay of time-critical IoV applications. The experiment demonstrates the proposed model can efficiently distribute the fog layer tasks while minimizing the delay.

Keywords: time-critical; fog computing; deep learning; internet of vehicles; task offloading

1. Introduction

Currently, with the development of machine learning, deep learning, computer vision, and 5G mobile communication technologies, the Internet of Things (IoT) techniques [1] have made tremendous progress and become a closely related part of people's lives. Consequently, the intelligent transportation technology represented by Intelligent Transportation Systems (ITS) and autonomous driving technology has achieved tremendous development due to a growing demand for smart cities and IoT technologies in modern society. As one of the most important popular applications of IoT technologies, the Internet of Vehicles (IoV) [2] technique has become an essential data transmission and resource scheduling framework in ITS and has attracted the attention of many researchers. Although the IoT and IoV technologies have become very hot research fields and achieved tremendous development, they have to face with some challenges because of their known limitations, such as restricted storage, real time critical, load-balancing, energy consumption, and so on. Artificial Intelligence (AI) [3] technologies such as Machine Learning (ML), Deep Learning [4] and deep neural networks, that are popular and have shown significant influence, are more and more applied in IoT and IoV fields and dramatically improve the effectiveness of IoT devices [5,6].

Generally, the IoVs are regard as data transmission platform that provide information exchange service between the vehicle, or vehicle and other surrounding devices through different communication media [7]. Through deep integration with the ITs, the IoV builds an intelligent network to provide essential functions for transportation systems,such as intelligent traffic management, dynamic information services, intelligent vehicle control, among others [8]. The architecture of IoV shown in Figure 1 is composed of three fundamental components: the inter-vehicular network (V2V), intra-vehicular network (V2I), and vehicular mobile Internet. Every vehicle in the IoV is connected with other vehicles and devices through Mobile Internet all time. The IoV creates an interconnected network for all vehicles to enable the exchange of information passengers, drivers, sensors and electric actuators, and the Internet by using advanced communication technique, such as IEEE 802.11p, cellular data networks (4G/5G) directional medium access control (DMAC), vehicular cooperative media access control (VC-MAC), and others.

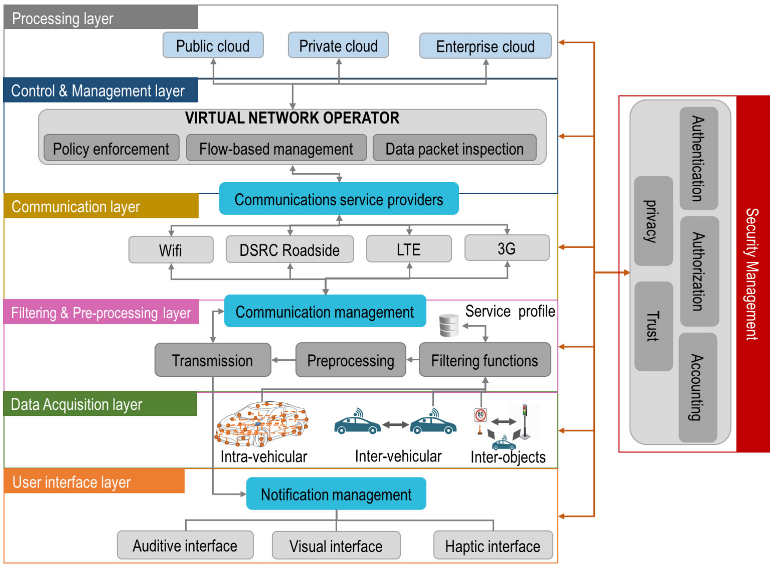


Figure 1. The IoV layered architectures in [7]

However, this early proposed IoV architecture faces with a challenge of real-time critical requirement. Specifically, the IoV may be susceptible to significant latency when storing or retrieving data, as when multiple vehicles access data simultaneously, it is limited by the data storage and processing capabilities of the vehicular cloud layer, resulting in significant latency. Under this situation, the problem of network congestion often appear in IoV networks, which can cause many issues that influence the operation of network, such as reduction of QoS and long time delay in data transmission [9]. From the view of this point, therefore, the IoV networks are time-critical systems [1,10].

The main challenge of time-critical systems is to ensure that tasks with real-time constraints in the system meet their respective deadlines. This problem presents an astonishing number of challenges and research opportunities for modern computing systems, which are not specifically designed to support time criticality [10]. However, the significant latency is the main challenge of current IoV architecture. In cloud computing-based IoV architecture, the data sources are often far away data process and store server, which are main reason to cause long time latency and lead to slow response times [11]. Therefore, these features limit the application of these frameworks to apply in many cases with less stringent functional requirements of real-time or timely intervention, thereby limiting the scope of vehicle services that cloud-based IoT frameworks may provide [11–13]. In responding to these concerns and limitations, the Edge and Fog Computing based IoV application frameworks offering better response time and privacy preservation [14] by moving data process and store to fog or edge layer to reduce distances and significant latency. In building Edge/Fog Computing infrastructures, the popular AI techniques including ML, DL, or Reinforcement Learning (RL) [15] algorithms are widely applied, which makes the intelligent data processing possible at the edge of network. This novel technique of edge/fog computing greatly reduces the application latency and improves the privacy offered to each vehicle.

Based on the fog computing, the reference of [14] proposed a new paradigm for IoV called Vehicular Fog Computing (VFC), in which the end-to-end latency is deeply investigated. Based on early architecture shown in Figure 1, a special layer called fog computing is introduced to reduces the delays and improves QoS. The fog layer consists of a large number of interconnected fog nodes which are data processing servers. Many IoV cloud-like services are provided by fog nodes using FC technique. However, the study of VFC in IoV is still in its early stage and there are several problem required to research deeply, such as congestion avoidance, guaranteed end-to-end delay, resources and tasks offloading, fault tolerance, security, and so on [16,17].

To reduce time delay and reach time-crucial requirement, the proposed architectures [18–21] applied some effective methods by reducing energy consumption, end-to-end delay, and communication resources in IoV networks. However, there are several important issues such as dynamic computational costs for load balancing and minimizing IoV networks' delay, and dynamic IoV topologies that are still at early stage of research, and required to study deeply [14].

Focusing on these issues, we propose a new IoV architecture in this paper by combining the advantages of AI based time-critical system, deep learning approaches, and edge(fog)/ cloud-based IoT technologies. Benefiting from the advantages of AI and fog computing-based vehicle network, the proposed architecture guarantees reliable and low latency communication in a highly dynamic environment. As the edge node, the processers in each vehicle need process large data collected from sensors and implement many task. In this paper, we propose a task allocation and offloading algorithm based on the SDV-F framework by applying deep learning technique to distribute tasks and computation resource efficiently and minimize the end-to-end delay.

The main contributions of this article are as follows:

- We propose a novel AI-based architecture for IoV network based on SDV-F framework, which can help to minimize end-to-end delay in data transmission.
- We propose an AI-based time-critical task allocation approach in IoV network, in which the AI algorithms such as DL and RL are applied to implement task offloading and resource allocation.
- We propose deep network-based Reinforcement Learning framework for resource allocation and task offloading approach in IoV network.

The rest of this paper is organized as follows. Section 2 introduces the background and challenges of this study. Section 3 describes the framework of the proposed model. The evaluation criterion simulation results are presented in Section 4 and, finally, Section 5 gives the conclusions.

2. Background and Challenging

Over the last decade, many researchers have presented various architectural configurations for IoV service. The main targets of these paradigms are to reduce end-to-end delay by applying advanced technologies or proposing novel architectures.

The early IoV service architecture is proposed in [22,23], the basic architecture of which is shown in Figure 1. In [22,23], the researchers combined IoT-aware technique with ML and cloud computing techniques for IoV architecture. In this kind of early architectures, the devices of vehicles accessed to connect internet by the managements from anywhere send data to a cloud computing server which implements data processing, storage, transmission, and other facilities. To ensure seamless connectivity, the communication layer needs to apply advanced wireless communication technologies such as GSM, WiFi, 3G/4G mobile network, or others. Although various technological alternatives have been adopted to meet specific needs or application scenarios and improve the efficiency and reliability of the cloud-based IoV infrastructure, the limitations of cloud-based architecture configuration are privacy and latency issues. These were mainly related to the centralized cloud server location and network infrastructure for communication and data transmission. As a result, on responding to the requirement of speed, accuracy, and reliability, this kind of approaches can not achieve satisfied time-critical solution.

As the mobile communication networks used in SDV-F are more and more intelligent and efficient, one of effective way to solve the shortcomings of time-critical system is to apply advanced data transmitting technologies or proposed efficient IoV architectures or approaches. With the development and popular application of 5G cellular mobile communication, Huang et.al [24] proposed a 5G-enabled SDV Network (5G-SDVNs) to provide communication service in IoV. In improving performance of data transmission in dynamic vehicular networking environments, the reference of [25] proposes novel approach which is under the framework of SDN-based Medium Access Control (MAC) protocol. At the same time, further work is being carried out in [26], in which the authors proposed the MCH framework. This MCH (Mobile Cloud Hybrid) framework are often applied to decrease the power consumption of mobile terminal or robotics. At the same time, Chen et.al. [18] proposed another cloud computing framework for mobile systems framework. By mean of these approaches, each mobile user's independent task can be processed locally in the Computing Access Point or on a remote cloud server.

The emerging autonomous driving technology for vehicles requires many applications, such as real-time situational awareness, collaborative lane changing, and vehicle stability control. These applications need edge computing technology to provide sufficient computing resources at edge nodes to perform time critical and data intensive activities. To meet these requirements, Zhu et.al. [27] did deeply research on the latency issues and task allocation in vehicular fog computing and proposed a novel approach called Folo. Moreover, the Folo approach is improved to support vehicles' mobility by creating edge computing layer for IoV, in which the producing tasks acting are as fog nodes. However, the optimization in Folo is an NP-hard problem, which is approximately solved by applying the Linear Programming technique only.

Applying fog computing is a great improvement to decrease the time delay in data processing and transmission. The fog nodes are designed closer to the bottom nodes, i.e. vehicles, to process and transmit the data produced by vehicles in the fog computing-based IoV network,. The research work of [5,28] investigated the applications of SDN on large-scale usage of Vehicle Networks (VN) services base on fog computing framework. The study on management of each wireless network composing VN is still in its early stage, however, which limits the development of fog infrastructures.

Undoubtedly, some recent research work offer good models or approaches to minimize latency, build time-critical system to manage the task offloading issues in the SDV-F architecture. However, most of these works are limited in using the multi-agent system and the horizontal fog layer resource pooling, which are demonstrated to substantially decrease the data respond latency [5]. Focusing on the time-critical computation applied in VSDN of IoV service architectures, in this article, we present AI-based hierarchical framework for SDN-F. We propose a AI-based time-critical system to manage task allocation and offloading to fit real-time requirement. We also present a AI (ML,DL) based fog computing network supporting between fog nodes, vehicles-to-fog nodes, and fog layer to cloud layer tasks and traffic offloading to attain intelligent resource allocation and minimize the end-to-end latency.

3. Proposed System Architecture

Based on the analysis reported in the literature review and inspired by the solutions proposed in the previous papers, in this section, we present the architecture for a IoV service system that incorporates advanced AI-based time-critical technologies, fog computing, and deep learning approaches. The proposed architecture contains three main layers: Intelligent Data Acquisition Layer or IoV Layer, Fog Computing Layer, and Data Visualisation and AI-based SDN Controller Layer, as illustrated in Figure 2.

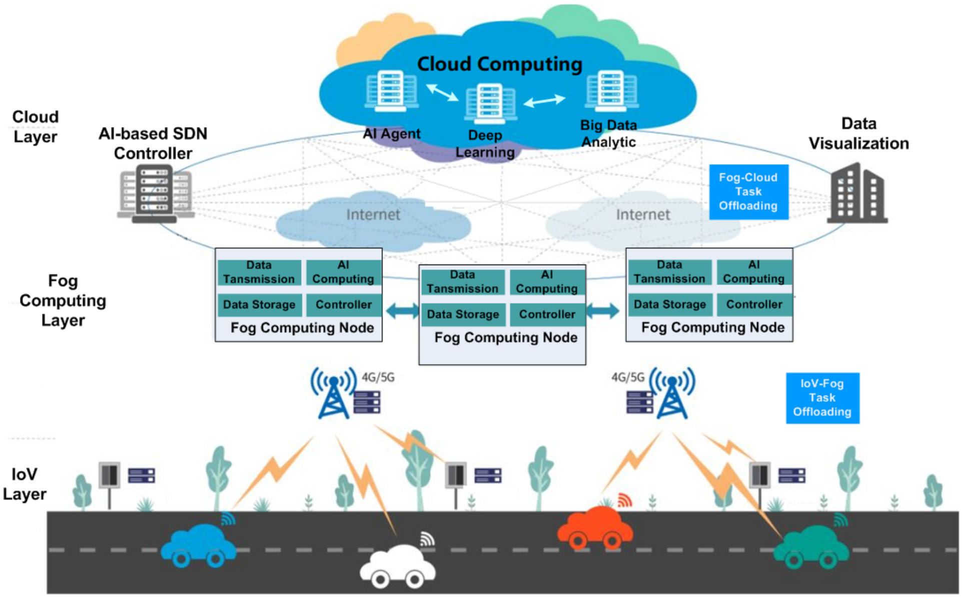


Figure 2. The base architecture of proposed system.

3.1. Layers of System Architecture

The Intelligent Data Acquisition Layer is also called IoV layer which includes a large number of IoV devices. Each vehicle contains a complex computer system which processes large scale of data from many sensors, implements and allocates multiple tasks. The vehicles are edge-nodes of edge computing network to communicate with Base Station (BS) by using 4G/5G mobile communication.

The fog computing layer is a fog computing networks which consists of many fog computing servers providing network communications, data storages, data processing and computing to IoV devices. These many servers are also called fog nodes. In real-world applications, the vehicles in motion will generate a large scale of data representing their real-time status all time. The mainly function of the fog nodes is to process and upload so large scale of data to the control servers. Moreover, these implementations of fog nodes have to meet the requirements of real-time critical and low latency. Obviously, it is very difficult for fog nodes to complete these tasks successfully. Therefore, it is significant important for IoV system to enable to perform distributed computing and implement a load balancing technique to control load and reduce latency. More deeply, besides fog computing-based network architecture, the efficient AI-based algorithm for task and resource offloading is also essential. Other target of this paper is to propose task and resource offloading approach by applying AI algorithms.

The high-lever of the architecture is cloud computing layer which provides AI-based SDN controlling and data visualisation functions. In designing AI-based SDN controller, we adopt two-layer structure, i.e. the data process unit is separated from control unit. This structure helps to improve the evolution of the system and facilitates network management. The intelligent unit implements big data analysis and process, and make decision. The intelligent unit consists of three intelligent modules: intelligent agent module, big data analysis, and deep learning module. By taking into account the available computing resources and combining data analytic results provided by the big data analytic module, the deep learning module offers the best model for the fog node to execute on each fog node. By using the intelligent techniques, the AI unit can make intelligent decisions adaptively.

3.2. Intelligent Data Acquisition Layer (IoV Layer)

The Intelligent Data Acquisition Layer includes a large number of IoV devices. Each node of IoV layer is a complex computer system which is also divided into three layers: advanced sensors and sub-system layer; computational, storage and process unit, and Artificial Intelligence module, as shown in Figure 3.

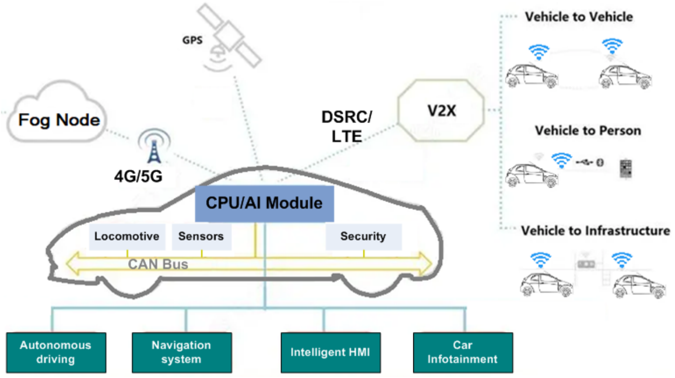


Figure 3. The illustration of the Intelligent Data Acquisition Layer.

3.2.1. AI-based Task-Allocation Algorithm in IoV Node

The advanced sensing technologies are used to collect data related to real-time status of vehicles on motion. The multiple-sensor techniques allow the system to collect important correlating data that can be used by AI module to make meaningful decision ensuring IoV frameworks more robust and trustworthy. The computation and data processing unit is key part in this system. To achieve real-time response and forwarding of processed data to upper layers, it is necessary to equip AI-based algorithm to central processing unit (CPU). By applying AI techniques such as deep learning, machine learning, or reinforcement learning, the AI based algorithms can make intelligent decisions based on data analysis. In this subsection, we propose a Reinforcement Learning (RL) [29] and deep neural network (DNN) based algorithm to fulfil task allocation with time critical-aware.

The CPU is central processing unit which manages many tasks of multiple sub-systems and makes decisions to provide distributed and efficient resource management. The responding algorithm is based on the framework of Markov Decision Process (MDP) [30] and RL and embedding deep neural network to enable the servers make effective decisions adaptively.

3.2.2. MDP-based Reinforcement Learning

In this on board system, we assume the central CPU as main agent, and other sub-systems as general agent. In RL algorithm, three components are needed: States, Actions, and Rewards.

According to the principle of RL shown in Figure 4, the CPU of servers are regard as agents (i.e., the central CPU is the primary agent) output actions to the environment based on perceived states. The environment represents the task allocation or offloading system, which evaluates the current actions and outputs the reward function and states. Based on the evaluation, a value function related to reward of actions records the value difference between the current and previous state-action pairs. Consequently, the long-term rewards represent the total rewards that the CPU (i.e., the agents) can expect to accumulate over time for each environmental state. Following this process, the RL model provides a long-term value about the future states based on their corresponding rewards. With the previous reward and value function, finally, the system model evaluates the current action to optimize a best reward and value of the next state.

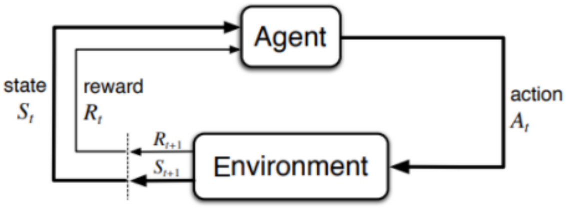


Figure 4. The framework of Reinforcement Learning.

The theory of the Markov decision processes provides mathematical foundation for the proposed system. We represent Markov decision processes function with a tuple of (s, a, P, r), where $s \in S$, $a \in A$, and $r \in R$. The symbols of S, A, and R represent the set of states, actions, and rewards. Additionally, the symbol of P is the transition probability, which represents the probability of cyclic process that the current state s produces the next state $s^{'}$ under the condition of current action. The value of the probability P is between 0 and 1. Accordingly, the $P(s^{'} \mid s, a)$ is probability of a new state of the environment. This new state is generated under the environment that is represented with state s and the chosen action a. The $R(s, a, s^{'})$ is the reward function of new state with the current state, that is generated by environment after the action . The reward functions' value represents with discount factor γ, where the value of the discount factor is $0 < \gamma < 1$.

Action Selection Policy

The policy is a necessary component in RL which defines the behavior of agents. A policy π is a distribution over action a given state s:

$$\pi(a \mid s) = P(A_t = a \mid S_t = s) \tag{1}$$

In RL, an agent attempts to seek optimal policy π that the agent achieves maximized the sum of rewards that is called utility. The utility function can be represent as follows:

$$U_h([s_0, s_1, \ldots, s_n]) = \sum_{i=0}^{n} \gamma^i R(s_i) \tag{2}$$

where γ is discount factor and $R(\cdot)$ is reward function.

State-Action Quality Function

In MDP, the dynamic programming (DP) technology is applied to solve P and R. The DP is a optimization which seeks best choices by using a optimal value function. To realize the RL by using MDP model, there are three functions required to optimized which are he value of state (s), $U^*(s)$, and $Q^*(s, a)$. The $Q(s_t, a_t)$ represents the value of action that the agent takes at the current state s_t. According to the principle of RL shown in Figure 4, the agent requires to choose an new action for the current state s_t based on the rewards generated by environment. Specifically, before selecting a new action, the agent computing $Q(s_t, a_t)$ for each possible action and then decides the new action of the current state s_t according to the optimal policy. In optimization, $\pi^*(s)$ is a optimal policy defined as optimal action selection from state s to a new state $s^{'}$. In this paper, we apply the Bellman Equations as optimizing target [14], in other word, the optimum action has to satisfy the Bellman Equations Eq.(3).

$$U_h^*(s) = \max_{a \in A(s)} Q^*(s_t, a_t) \tag{3}$$

where

$$Q^*(s_t, a_t) = \sum_{t=0}^{T} P(s^{'}|s_t, a_t)\left[R(s^{'}, a_t) + \gamma^t U_h^*(s^{'}, a_t)\right] \tag{4}$$

and

$$U_h^*(s_t, a_t) = \max_{a \in A(s)} \sum_{t=0}^{T} P(s^{'}|s_t, a_t)\left[R(s^{'}|s_t, a_t) + \gamma^t U^*(s^{'}, a_t)\right] \tag{5}$$

where (s_t, a_t) represents the state-action quality pair at time t, and T is the time limit of the agents' optimization problem in the proposed model.

3.2.3. Deep Q-function Learning for Task Allocation

On the base of RL, we apply DQN (deep Q-function network)) to predict the value of (s_t, a_t). In this model, the deep neural network (DNN) is used to learn the agent's Q-function $Q(s,a)$.

$$Q^*_\lambda(s_t,a_t) = E\left[R(s_t,a_t) + \gamma \max_{a^{'} \in A_{s^{'}}} Q^*_\lambda(s^{'},a^{'})\right] \tag{6}$$

where $(s^{'}, a^{'})$ is the state-action pair at the next time slot, and $A_{s'}$ is the set of actions at the next state $s^{'}$. The Figure 5 illustrates the basic framework of DQN, in which we use convolutional neural network (CNN) as DNN. The symbol of λ represents the parameter DNN.

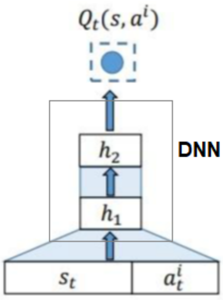


Figure 5. The illustration of the Intelligent Data Acquisition Layer.

In training model, the mean-squared error (MSE) is applied as loss function of the proposed model with the parameter CNN λ, which is defined as follows:

$$Lo(\lambda) = E\left[(q - Q_\lambda(s,a))^2\right] \tag{7}$$

where

$$q = R(s,a) + \gamma \max_{a^{'} \in A_{s^{'}}} Q^*(s^{'},a^{'}) \tag{8}$$

is the maximal summary of future reward for the agents task allocation process.

3.3. Fog Computing Layer

The Fog Computing Layer consists of a large number of fog nodes, which frequently process and upload real-time data generated by vehicles on motion to the SDN-controller. To control the tasks load and reduce latency, the officiant algorithm is vital and essential to realize distributed computing and load balancing technique.

3.3.1. AI-based Task-Offloading Algorithm in Fog Node Layer

The proposed task-offloading algorithm is also based on RL framework introduced in pre-subsection [14]. But we improve the algorithm by proposing novel architecture of DNN network, reward function, and Q-function.

Task-Offloading Model for Fog Nodes

The target of the approach is to optimize the offloading operations of each agent to achieve maximum utility under the condition of minimizing time latency and optimizing the allocation of IoV tasks. Therefore, we apply the reward function is defined in [14]:

$$R(s,a) = U(s,a) - (P_l(s,a) + D_L(s,a)) \tag{9}$$

where $P_l(s,a)$ represents the traffic load probability function of a fog node f_j, and $D_L(s,a)$ indicates the end-to-end delay function. Unlike Eq.(2), the utility function is defined as:

$$U(s,a) = r_u log(1 + t_o) \tag{10}$$

where t_o is the number of tasks offloaded to fog node f_j, and r_u is the utility reward.

We apply the reward function Eq.(9) to get the appropriate reward value of the selected fog node for task computation and a next state s' in this proposed algorithm. Since the arrives of next fog node with tasks to be assigned and the task size after each node's task are random, the Poisson random variables are used in the proposed model [31].

Following the Eq.(9), the probability function $P_l(s,a)$ of a fog node f_n is required to compute as follows:

$$P_l(s,a) = W_t \frac{t_c P_c + t_o P_o}{t_c + t_o} \tag{11}$$

where the probability of P_i $(i = c, o))$ is modeled by a Poisson process and with the follows:

$$P_i = \frac{\max(0, r_{ar} - (\max(Q_i) - Q_i^{'}))}{r_{ar}} \tag{12}$$

where W_t represents the weight of traffic load, t_c is the currently processing tasks, r_{ar} indicates the tasks' arrival rate at fog node f_i. The symbol of $Q_j^{'}$ represents the next estimated queue state of a fog node f_i with a given state s and action a,

$$Q_i^{'}(s,a) = \min(\max(0, Q_i) + k_i, \quad \max(Q_i)) \tag{13}$$

The end-to-end delay $D_L(s,a)$ of task is very important in the proposed model. We compute it using the following equation.

$$D_L(s,a) = W_d \frac{td_e + td_q + td_t}{t_p + t_o} \tag{14}$$

where, W_d is the delay weight, td_e is operation delay, td_q is time delay of queue, and td_t is time delay of data transmission delay. In the proposed model, td_q represents the waiting time of the current node f_i in the queue, and td_e depends on the running-speed of processor in f_i.

Optimizing Task-Offloading Algorithm

Due to the dynamic nature of the IoV network, it is hard for controller to predict the R and P. Based on the fact that the reward and probability distribution are stable, we also apply DNN based RL technique. The base frame work is also shown in Figure 5. We use a U-Network (U-Net) [4] shown in Figure 6 as DNN to learn Q-function $Q(s,a)$.

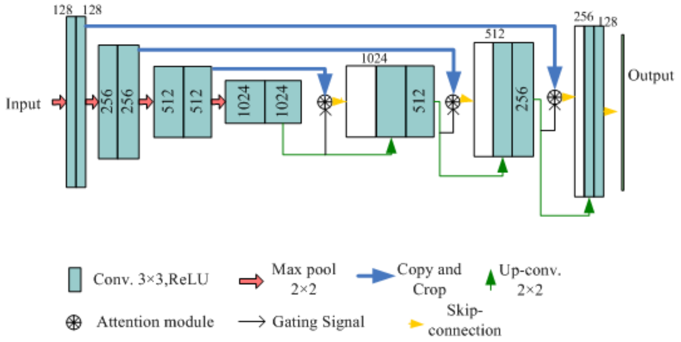


Figure 6. The architecture of U-Net applied in DQN.

The architecture of U-Net is an autoencoder with attention module, which can make learning process focus on important agents quickly and maximize the foreseen reward function. The architecture of network consists of three downsampling layers, each of which includes two convolutional layers and a max-pooling layer with 2 × 2 pooling. Accordingly, there are three upsampling layers at the other side of network. Each of upsampling layer consists of two convolutional layers with 2 × 2 upsampling. Before input each upsampling layer, the attention module is applied to fusion and calculate a single scalar attention value. To reduce the delay and fit to real-time requirement, we use 1 × 1 convolutional kernel in the attention module as shown in Figure 7.

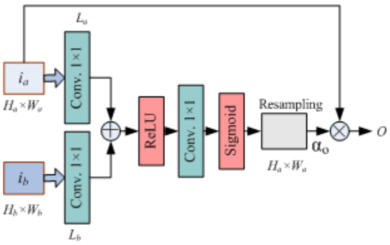


Figure 7. The attention module of U-Net.

In optimization, the target of the model is to select a optimal policy of (s, a) in the system. Specifically, based on current policy (s, a), the system foresees the new state $s^{'}$ and the reward by using Q-learning. The Q-learning is a continuous optimization in which the Q-function is updated with each of iterations to make the best decision for the new task. The updating equation of Q-function is as follows:

$$Q^{*}(s, a) = (1 - \delta)Q^{*}(s, a) + \delta\left(R(s, a) + \gamma \max_{a^{'} \in A_{s^{'}}} Q^{*}(s^{'}, a^{'})\right) \tag{15}$$

where δ is the learning rate which is a factor between 0 and 1 $(0 < \delta < 1)$, and γ is the discount factor. In optimization, the reward function R is modified based on the new learning rate.

4. Experiments

4.1. Experiment Setup

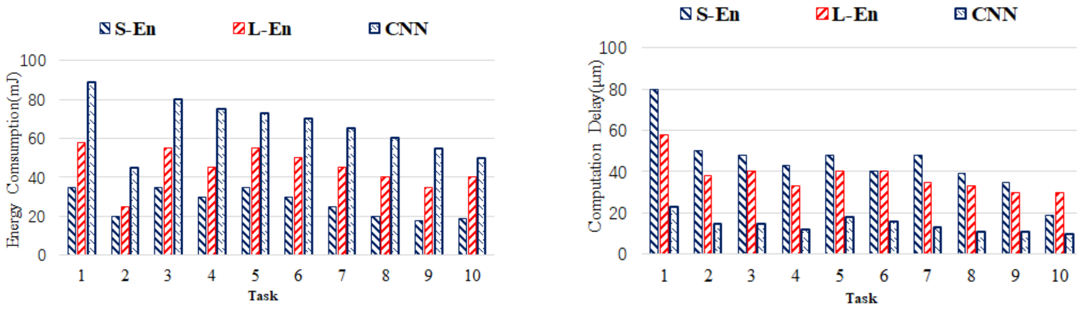
To evaluate the optimization performance, we simulate a macro IoV environment based on SDV-F framework covering the area with about 0.5 × 0.5 km^2. We set 5 RSUs and BS as fog nodes in the simulation network, and we use five cars as moving vehicles with speed from 30 to 100 km/h. We set the parameters of transmission as follows: the CPU size for each content is $Rcpu = 1$ cycle/bit, the power of transmission is 100 mW, the maximal delay is 0.2 second, and the bandwidth is 100MB/s.

We will first discuss simulation results of task allocation of IoV nodes, i.e. task process system of CPU in each vehicle. We do this experiment in laboratory by using an embedded experimental platform. To evaluate the performance of the proposed network, we consider three baseline models: ARTNet [14], eDors [32],and the Energy-Constrained Signaling Reduction model (ECSR) [33]. The ARTNet is an AI-based resource allocation and task Offloading model for IoV networks, providing reliable and short delay communications. The eDors is an energy-based offloading scheme.It combines resource optimization and dynamic offloading to reduce the minimization of full-time collaborative applications. The ECSR model built an average restriction on RSU in each time slot to obey the long-term energy constraint.

4.2. Results of Task-Allocation Model

As introduced in subsection 3.2, the proposed task-allocation model uses DNN as Q-learning architecture. To investigate the performance in deferent neural network, we design three architectures of network: a large-scale autoencoder, a small-scale autoencoder and a general CNN. The large-scale autoencoder (L-En) consists of four hidden layers and each of layer includes more nodes than small scale autoencoder. The architecture of it is 256 × 512 × 256 × 128. The small-scale autoencoder (S-En) consists of only two hidden-layer with architecture of 64 × 128. The CNN consists of two convolutional layer and a full-connected layer. In convolutional layer, we apply 3 × 3 filter kernel and max-pooling technique. In the simulations, we compared the energy consumption and computation delay with numbers of tasks.

The Figure 8 shows the comparison of energy consumption and computation delay for different DNN architecture in Q-learning process. From the figure we can find out that for a given task, the energy consumption increase as the architecture becomes more complex, however, the changes of the computation delay are opposite. The reason about this is that the DNN with more complex architecture consists of more nodes and requires more computation, but it can provide more learning performance, that leads to small time delay. We also find a more interesting conclusion from Figure 8 that with the increase of number of tasks at the same time the task-allocation model decreases the energy consumption and computation delay. This is because the AI-based algorithm makes reasonable optimization arrangements based on the increase in task count.



(a) Energy consumption

(b) Computation delay

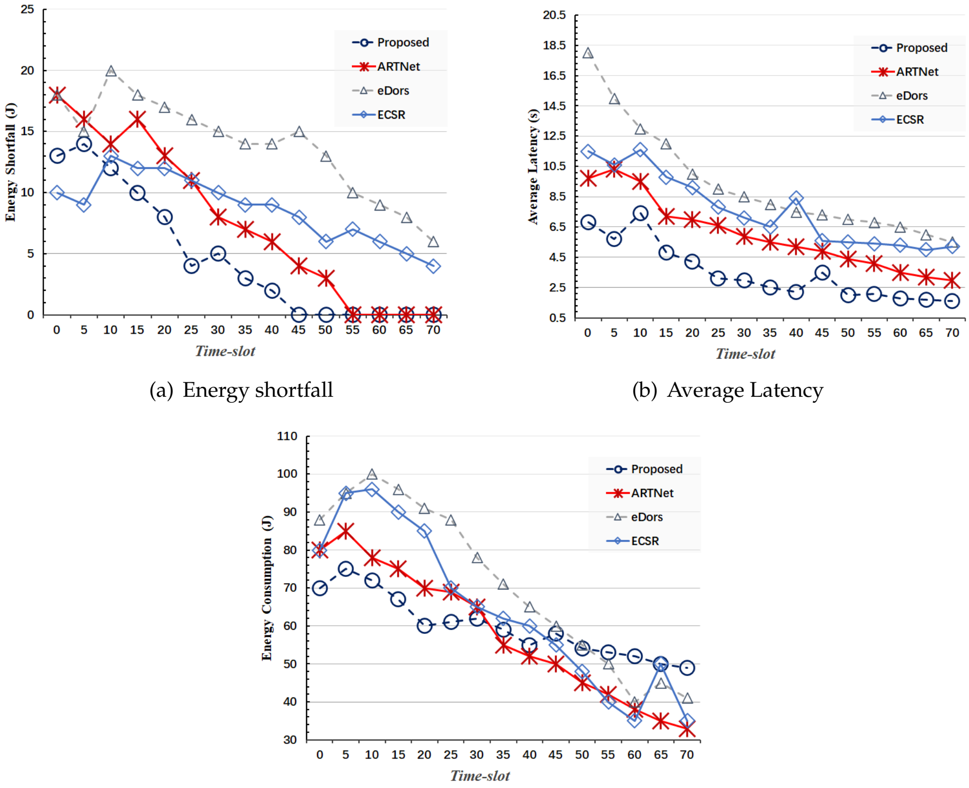
Figure 8. Comparison of energy consumption and computation delay for different DNN architectures.

4.3. Results of Proposed Task-Offloading Model

Similar with [14], we evaluate the performance of proposed model in following aside: performance under different time-slots, the number of IoVs, and vehicles with different speed.

4.3.1. Performance under different time-slots

The comparison results are shown in Figure 9. By comprehensively analyzing these three figures, we can draw conclusion that the performance of our proposed model is much better than other baselines.



(a) Energy shortfall

(b) Average Latency

(c) Energy consumption

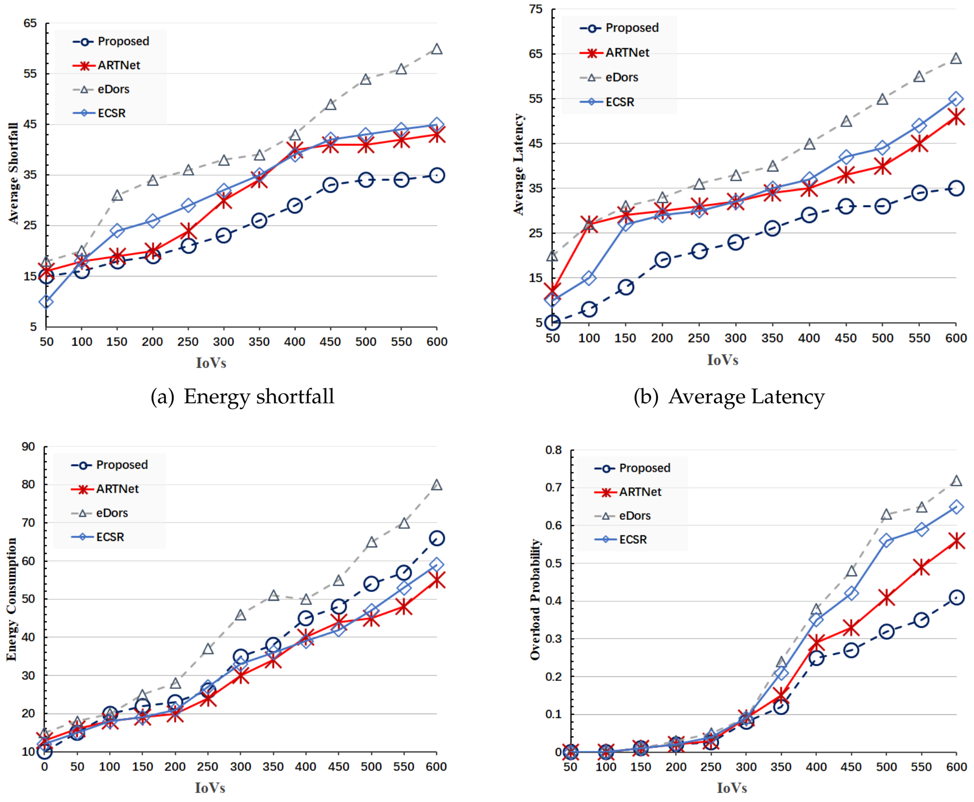
Figure 9. Comparison of performance of proposed model and other baselines in different time-slot.

The Figure 9(a) shows the results of energy shortfall for all models. From this sub-figure, overall, this model is with better performance. At first, the ECSR is with less energy shortfall than other models, but with the time-slot more than 15, our proposed model is with less energy shortfall than the ECSR. The performance of the ARTNet is good, since the energy shortfall is zero when the time-slot increase more than 55. From the figure, the energy shortfall is less than the ARTNet all the time and it decrease zero after 45 of time-slot. Conclusively, the performance is much better than that of the ARTNet. This is because our proposed model apply AI technique in architecture.

From the Figure 9(b), we find that the average latency is much less than other baselines. In communication system, the average latency is the most important in evaluation. The results from this figure demonstrates the outstanding performance of our proposed model. However, by analyzing energy consumption results, our proposed model does not have advantage. When time-slot increases more than 30, our proposed model requires the most energy consumption. The reason of it is that our proposed model uses DNN neural network in designing algorithm, which makes the processor require more computation and energy consumption. Conclusively, although the proposed model consumes more energy, it is succeed in terms of lower latency and lower energy shortfall, which is benefitted of its intelligent distribution of tasks offloading algorithm at the fog layers.

4.3.2. Performance under different number of IoV nodes

To evaluate the carrying capacity of the proposed model, we implement it and other compared model in the contests of various number of IoVs. In this simulations, we design a software to simulate many IoVs by sending tasks requirement to IoV network. We evaluate the model on the following aside: latency, energy consumption, energy shortfall, and overload probability. The comparison results are trend that varies with the number of IoVs. The comparison results are shown in Figure 10.



(a) Energy shortfall

(b) Average Latency

(c) Energy consumption

(d) Average Overload Probability

Figure 10. Comparison of performance of proposed model and other baselines in different number of IoVs.

From the results shown in Figure 10, we can find out that our proposed model achieves better performance in terms of average latency, average energy shortfall, and average overload probability when the IoVs varies from 50 to 600 in the simulation. But it gets similar results in energy consumption. The average energy consumption of all compared models increase linearly with the increasing of IoVs. From the results, the average energy consumption in the proposed model is more than ECSR and ARTNet. The reason of it is related with the complexity of deep network in the proposed model. When the architecture of proposed model includes more layers and nodes, it requires more computation and energy consumption. From the viewpoints of time-crucial requirement, the average latency is the most important target in a time-crucial system. As shown in Figure 10(b), the proposed model achieves the best performance in average latency, i.e, the latency is the lest. At the same time, the proposed model gets fewer average shortfall than other compared models, when the number of IoVs increases more than 250. The shortfall is a performance measure to evaluate the stability of system operation. The less the number of shortfall is, the more stable the system runs. Therefore, the Figure 10(a) indicates the proposed model achieves outstanding performance in system running. The average overload probability is shown in Figure 10(d). When the number of IoVs is more than 500, the overload probability in eDors and ECSR is more than 50%, and in ARTNet, it is more than 40%. At the same time, the average overload probability in the proposed model is much less than other models. In other word, the proposed model distributes the tasks efficiently.

Conclusively, the proposed model effectively decrease the latency without excessively increasing average energy consumption. The AI-based task allocations in IoV node increases the performance of the proposed IoV network.

5. Conclusions

In this paper, we proposed a novel framework for IoV network by considering the application and requirement of time-critical system. Based on the requirement of time-critical application, we first study the problems of reliable low-latency communication and tasks offloading in dynamic environments in IoVs network. Focusing on these problems, we proposed an AI-based tasks and resource offloading model for IoVs network, which ensures reliable low-latency communication efficient tasks offloading in IoV network by using SDV-F architecture. By applying AI technologies of RL, Markov decision process, and deep learning, the proposed model intelligently distributes the fog layer's traffic load according to the computational power and load on each fog node. By proposing a AI-based task-allocation algorithm in IoV layer, the proposed model effectively reduces unnecessary task allocation at the fog computing layer, thereby improving the efficiency of the distribution of tasks and resources and then reducing time delay.

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflict of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript, or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results".

References

1. Jan, M.A.; Zakarya, M.; Khan, M.; Mastorakis, S.; Menon, V.G.; Balasubramanian, V.; Rehman, A.U. An AI-enabled lightweight data fusion and load optimization approach for Internet of Things. *Future Generation Computer Systems* **2021**, *122*, 40–51. doi:https://doi.org/10.1016/j.future.2021.03.020.
2. Wang, X.; Ning, Z.; Hu, X.; Wang, L.; Guo, L.; Hu, B.; Wu, X. Future communications and energy management in the Internet of vehicles: Toward intelligent energy-harvesting. *IEEE Wireless Communications* **2019**, *26*, 87–93.
3. Yadav, S.P.; Mahato, D.P.; Linh, N.T.D. *Distributed artificial intelligence: A modern approach*; CRC Press, 2020.
4. Liang, P.; Liu, G.; Xiong, Z.; Fan, H.; Zhu, H.; Zhang, X. A facial geometry based detection model for face manipulation using CNN-LSTM architecture. *Information Sciences* **2023**, *633*, 370–383. https://doi.org/10.1016/j.ins.2023.03.079.
5. Ibrar, M.; Wang, L.; Muntean, G.; Chen, J.; Shah, N.; Akbar, A. IHSF: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems. *IEEE Internet of Things Journal* **2020**, *8*, 3130–3142.
6. Liang, P.; Liu, G.; Xiong, Z.; Fan, H.; Zhu, H.; Zhang, X. A fault detection model for edge computing security using imbalanced classification. *Journal of Systems Architecture* **2022**, *133*, 102779. doi:https://doi.org/10.1016/j.sysarc.2022.102779https://doi.org/10.1016/j.sysarc.2022.102779.
7. Contreras-Castillo, J.; Zeadally, S.; Guerrero-Ibañez, J.A. Internet of vehicles: architecture, protocols, and security. *IEEE internet of things Journal* **2017**, *5*, 3701–3709.
8. Guerrero-Ibanez, J.A.; Zeadally, S.; Contreras-Castillo, J. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications* **2015**, *22*, 122–128.
9. Mukherjee, M.; Matam, R.; Shu, L.; Maglaras, L.; Ferrag, M.A.; Choudhury, N.; Kumar, V. Security and privacy in fog computing: Challenges. *IEEE Access* **2017**, *5*, 19293–19304.
10. Mitra, T.; Teich, J.; Thiele, L. Time-critical systems design: A survey. *IEEE Design & Test* **2018**, *35*, 8–26.
11. Shumba, A.; Montanaro, T.; Sergi, I.; Fachechi, L.; De Vittorio, M.; Patrono, L. Leveraging IOT-aware technologies and AI techniques for real-time critical healthcare applications. *Sensors* **2022**, *22*, 7675.
12. Merenda, M.; Porcaro, C.; Iero, D. Edge machine learning for ai-enabled iot devices: A review. *Sensors* **2020**, *20*, 2533.
13. Erhan, L.; Ndubuaku, M.; Di Mauro, M.; Song, W.; Chen, M.; Fortino, G.; Bagdasar, O.; Liotta, A. Smart anomaly detection in sensor systems: A multi-perspective review. *Information Fusion* **2021**, *67*, 64–79.

14. Ibrar, M.; Akbar, A.; Jan, S.R.U.; Jan, M.A.; Wang, L.; Song, H.; Shah, N. Artnet: Ai-based resource allocation and task offloading in a reconfigurable internet of vehicular networks. *IEEE Transactions on Network Science and Engineering* **2020**, *9*, 67–77.
15. Ling, C.; Jiang, J.; Wang, J.; Thai, M.T.; Xue, R.; Song, J.; Qiu, M.; Zhao, L. Deep graph representation learning and optimization for influence maximization. Proc. of International Conference on Machine Learning 2023. PMLR, 2023, pp. 21350–21361.
16. Kadhim, A.J.; Seno, S.A.H. Maximizing the utilization of fog computing in internet of vehicle using SDN. *IEEE Communications Letters* **2018**, *23*, 140–143.
17. Xiong, Z.; Li, X.; Zhang, X.; Zhu, S.; Xu, F.; Zhao, X.; Wu, Y.; Zeng, M. A service pricing-based two-stage incentive algorithm for socially aware networks. *Journal of Signal Processing Systems* **2022**, *94*, 1227–1242.
18. Chen, M.; Liang, B.; Dong, M. Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point. IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, 2017, pp. 1–9.
19. Whaiduzzaman, M.; Naveed, A.; Gani, A. MobiCoRE: Mobile Device Based Cloudlet Resource Enhancement for Optimal Task Response. *IEEE Transactions on Services Computing* **2018**, *PP*, 144–154.
20. Shuja, J.; Gani, A.; Ko, K.; So, K.; Mustafa, S.; Madani, S.A.; Khan, M.K. SIMDOM: A framework for SIMD instruction translation and offloading in heterogeneous mobile architectures. *Transactions on Emerging Telecommunications Technologies* **2018**, p. e3174.
21. Zeng, Y.; Pan, M.; Just, H.A.; Lyu, L.; Qiu, M.; Jia, R. Narcissus: A practical clean-label backdoor attack with limited information. *arXiv preprint arXiv:2204.05255* **2022**.
22. Miche, M.; Bohnert, T.M. The internet of vehicles or the second generation of telematic services. *ERCIM News* **2009**, *77*, 43–45.
23. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. Towards open-set identity preserving face synthesis. Proc. of the IEEE conference on computer vision and pattern recognition (CVPR, 2018, pp. 6713–6722.
24. Huang, X.; Yu, R.; Kang, J.; He, Y.; Zhang, Y. Exploring mobile edge computing for 5G-enabled software defined vehicular networks. *IEEE Wireless Communications* **2017**, *24*, 55–63.
25. Dai, P.; Liu, K.; Wu, X.; Yu, Z.; Xing, H.; Lee, V.C.S. Cooperative temporal data dissemination in SDN-based heterogeneous vehicular networks. *IEEE Internet of Things Journal* **2018**, *6*, 72–83.
26. Akbar, A.; Lewis, P.R. Towards the optimization of power and bandwidth consumption in mobile-cloud hybrid applications. 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2017, pp. 213–218.
27. Zhu, C.; Pastor, G.; Xiao, Y.; Li, Y.; Ylae-Jaeaeski, A. Fog following me: Latency and quality balanced task allocation in vehicular fog computing. 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2018, pp. 1–9.
28. Kadhim, A.J.; Seno, S.A.H. Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks. *Ad Hoc Networks* **2019**, *84*, 68–81.
29. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* **2017**, *34*, 26–38.
30. Puterman, M.L. Markov decision processes. *Handbooks in operations research and management science* **1990**, *2*, 331–434.
31. Kingman, J.F.C. *Poisson processes*; Vol. 3, Clarendon Press, 1992.
32. Guo, S.; Xiao, B.; Yang, Y.; Yang, Y. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing. Proc. of The 35th Annual IEEE International Conference on Computer Communications(INFOCOM 2016). IEEE, 2016, pp. 1–9.
33. Liao, Q.; Aziz, D. Modeling of mobility-aware RRC state transition for energy-constrained signaling reduction. Proc. 2016 IEEE Global Communications Conference (GLOBECOM). IEEE, 2016, pp. 1–7.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.