

## Article

# Federated-Learning-Based Energy-Efficient Load Balancing for UAV-Enabled MEC System in Vehicular Networks

Ayoungh Shin  and Yujin Lim \* 

Department of IT Engineering, Sookmyung Women's University, Seoul 04310, Republic of Korea

\* Correspondence: yujin91@sookmyung.ac.kr; Tel.: +82-02-2077-7305

**Abstract:** At present, with the intelligence that has been achieved in computer and communication technologies, vehicles can provide many convenient functions to users. However, it is difficult for a vehicle to deal with computationally intensive and latency-sensitive tasks occurring in the vehicle environment by itself. To this end, mobile edge computing (MEC) services have emerged. However, MEC servers (MECSs), which are fixed on the ground, cannot flexibly respond to temporal dynamics where tasks are temporarily increasing, such as commuting time. Therefore, research has examined the provision of edge services using additional unmanned aerial vehicles (UAV) with mobility. Since these UAVs have limited energy and computing power, it is more important to optimize energy efficiency through load balancing than it is for ground MEC servers (MECSs). Moreover, if only certain servers run out of energy, the service coverage of a MEC server (MECS) may be limited. Therefore, all UAV MEC servers (UAV MECSs) need to use energy evenly. Further, in a high-mobility vehicle environment, it is necessary to have effective task migration because the UAV MECS that provides services to the vehicle changes rapidly. Therefore, in this paper, a federated deep Q-network (DQN)-based task migration strategy that considers the load deviation and energy deviation among UAV MECSs is proposed. DQN is used to create a local model for migration optimization for each of the UAV MECSs, and federated learning creates a more effective global model based on the fact that it has common spatial features between adjacent regions. To evaluate the performance of the proposed strategy, the performance is analyzed in terms of delay constraint satisfaction, load deviation, and energy deviation.



**Citation:** Shin, A.; Lim, Y. Federated-Learning-Based Energy-Efficient Load Balancing for UAV-Enabled MEC System in Vehicular Networks. *Energies* **2023**, *16*, 2486. <https://doi.org/10.3390/en16052486>

Academic Editor: Mojgan Hojabri

Received: 26 January 2023

Revised: 3 March 2023

Accepted: 4 March 2023

Published: 6 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** mobile edge computing; unmanned aerial vehicle; task migration; deep reinforcement learning; federated learning

## 1. Introduction

At present, vehicles are becoming increasingly intelligent due to significant advancements that have been made in computers and communications. These advancements allow vehicles to use cameras, sensors, etc., to provide more convenient user functions, such as AR/VR, streaming service, or autonomous driving, but autonomous operation of these new applications requires enormous amounts of computing resources [1]. However, the vehicle's computing resources are not sufficient to internally handle all requirements. It is challenging to quickly handle computationally intensive and latency-sensitive tasks occurring in the vehicle environment. To address this, a powerful processor, such as a graphics processing unit (GPU), can be installed in the vehicle. However, this then requires greater power and cooling capabilities, leading to higher energy consumption, which ultimately significantly affects fuel economy and mileage [2].

Mobile edge computing (MEC) has emerged as a promising technology to overcome these limitations and efficiently perform tasks. The vehicle offloads tasks to the MEC server (MECS), and the MEC system can provide high quality of service (QoS) to the vehicle by providing fast and sufficient computing resources [3]. Moreover, unlike traditional centralized cloud computing, the MEC system has the advantage of reduced network

latency because it is geographically closer to the user. However, general MEC servers (MECSs) are fixed on the ground and have limited processing capacity. Therefore, it is impossible to flexibly respond to a sudden increase in task processing requests due to temporary congestion in traffic conditions, such as rush hour traffic. Moreover, installing additional ground MECSs in this scenario is not efficient in terms of resource utilization. Therefore, recent studies have considered mobile edge services that respond to temporarily increasing task requests by adding cheap and freely movable unmanned aerial vehicles (UAVs) [4,5].

However, the high mobility of users in a vehicle environment means that the vehicle can leave the MECS's service area before the MECS has completed the task. In addition, due to the movement of the UAV MECS, if the task is completed on the server but the vehicle is not present in a service area, then the computation result cannot be delivered. Moreover, irregularities in the occurrence of vehicles and tasks can lead to load imbalances between MECSs. Unlike MECSs that are fixed on the ground, UAV MECSs have limited battery capacity. Therefore, if too many tasks are concentrated on a particular server, only that server consumes energy quickly. Suppose the server runs out of energy completing all tasks offloaded to that server. In that case, the remaining tasks cannot be completed within the time limit, and the quality of service will be degraded. Even if the energy is not exhausted, the delay increases when the tasks are assigned to a specific server. To solve these problems, it is necessary to migrate the offloaded tasks in consideration of the load state and energy state of the UAV MECSs. Migrating tasks between UAV MECSs can resolve load imbalances and ensure that the energy of servers are used equally through load balancing between MECSs.

This paper proposes a task migration strategy between UAV MECSs that considers the UAVs load state and energy state in an environment where both the vehicles and the UAV MECSs are highly dynamic. An environment using UAV MECSs when the tasks are rapidly increasing and exceed the threshold of ground MECSs is assumed. Thus, ground MECSs that are already overloaded are not considered as task migration targets. We aim to reduce the load deviation and residual energy deviation of each UAV MECS and to maximize task throughput. A task migration strategy based on federated deep Q-Network (DQN) is proposed for three reasons: First, to achieve our goal in a highly dynamic environment where vehicles and UAV MECSs move often, a migration strategy must be flexibly adopted to respond to dynamic changes in the environment. Reinforcement learning (RL) is suitable for a dynamic environment because it can recognize the environment and make decisions through interactions with the environment [6,7]. Second, the efficiency of reinforcement learning decreases when the state and action space dimensions are large. DQN, a type of deep reinforcement learning (DRL), uses neural networks to predict the rewards of each action, and it is suitable for high-dimensional environments where the state and action space are large and complex. Finally, updating the DQN model through federated learning makes it possible to provide more efficient services suitable for spatial features. In other words, it is more effective to update the learning model by grouping regions with geographically common characteristics than it is to use different learning models for each region. Therefore, it is necessary to update the model through federated learning between adjacent areas with similar characteristics.

The main contributions of this study are as follows:

- (1) The system model is designed to use UAV MECSs together in environments where ground MECSs exist. Such a design can solve the overload of ground MECSs that can temporarily occur during the commuting time and reduce the processing delay time.
- (2) When migrating tasks, the energy deviation of the servers is also considered along with the total energy consumption of the UAV MECSs. In other words, the intention is to use the energy of all UAVs similarly while reducing the total energy consumption. This prevents defects within the entire scope of service provision. For example, if a UAV MECS's energy is exhausted, then service defects will occur within the service

range of this UAV. However, by considering energy usage deviations together, this situation can be prevented, and the load can be evenly balanced.

- (3) Additional UAV MECs are used to prepare for temporal situations when tasks increase rapidly. Federated learning balances the load to make it suitable for spatial characteristics. This makes it possible to provide the most efficient service in consideration of both the temporal and spatial characteristics of the downtown area during commuting times.

The remainder of this paper is organized as follows. Section 2 analyzes the related works, and Section 3 describes the system model and the problem formulation. Section 4 proposes a federated DQN-based task migration algorithm. Section 5 presents the simulation results, and Section 6 presents the conclusion and future research directions.

## 2. Related Works

There have been several studies examining algorithms for the load balancing of UAV MECs in vehicular networks. Recently, many task migration techniques have been proposed to effectively respond to dynamic vehicle environments. In this section, recently studied task migration techniques are introduced, and recent studies on migration strategies applied to systems with additional UAV MECs are presented.

Task migration in the vehicular environment is performed in consideration of the vehicle's direction of movement and with a main focus on minimizing delay and migration costs [8,9]. In a dynamic environment where vehicles are complex and fast moving, reinforcement learning that flexibly responds to environmental changes can achieve more effective migration strategies. In [10–12], delay and migration costs are considered, and tasks are migrated based on deep Q-learning (DQL), which is a type of reinforcement learning. They can maximize system utilization and dynamically adjust vehicle service migration decisions based on traffic information. This time, federated learning allows for the creation of a global model using data collected from each MEC server and updates each server's model. In [13], a reinforcement learning model that determines the migration of tasks in a vehicle edge computing environment updates through federated learning. This paper shows that federated learning in the vehicle environment has advantages, such as better efficiency and privacy, shorter response time, and better usability.

These migration algorithms can also be applied to UAV MEC environments. However, it is necessary to consider that UAV MEC servers have limited energy and computing power compared to grounded MECs, and this is a more dynamic environment with the movement of UAVs. Thus, reinforcement-learning-based strategies that consider the energy state and load state of UAVs are proposed. With such strategies, the energy state of the UAV accounts for flight energy, computational energy, and communication energy [14]. In [15], an energy-efficient dynamic task migration algorithm (EDTM) is proposed that minimizes the system's total energy consumption while ensuring the UAV system's load balancing. The proposed algorithm is based on an improved ant colony algorithm. That algorithm involved a path plan elimination strategy that comprehensively considers task migration distances between UAVs, the load situations of UAVs, and environmental factors in consideration of the UAV's load balance. In [16], it was assumed that the energy of the UAV was sufficiently supplied through solar energy. The researchers in that study also proposed an actor–critic reinforcement learning approach to learn near-optimal policy. They showed that this significantly reduced the task average response time.

Research on task migration strategies through federated learning in a UAV MEC-assisted vehicle environment has recently been conducted. In [17], a two-tier horizontal offload management (H-HOME) framework for migration in UAVs is proposed to minimize processing delays and jitter. That framework involved the usage of federated reinforcement learning to improve knowledge sharing without privacy and network overload problems.

The studies mentioned above attempted to reduce latency and increase throughput by adding UAV MEC to environments where ground MECs already exist. These studies only seek to reduce the total energy consumption of UAV MEC servers or assume that

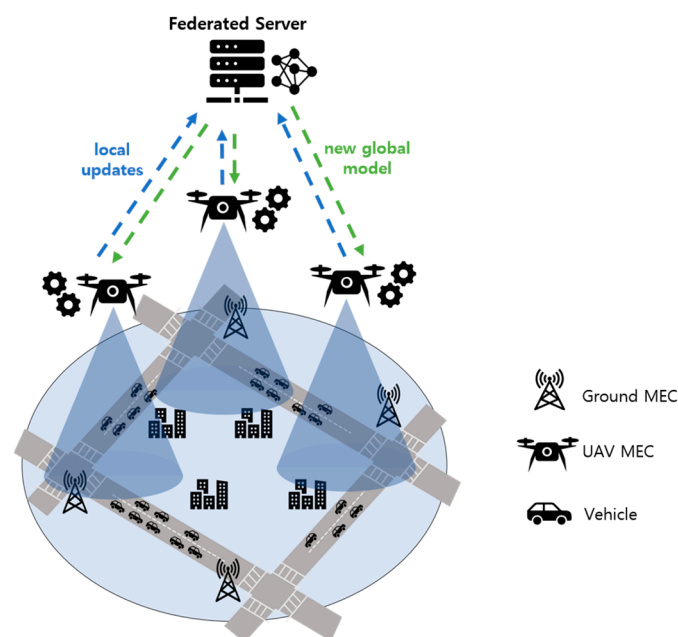
sufficient energy is supplied; they do not consider whether all UAV MEC servers use energy evenly. However, the even usage of energy is an important aspect of handling multiple tasks without exhausting all UAV MECs. Therefore, in this study, all-flight, computational, and migration energy of UAV MECs are considered, and a task migration strategy that can balance the load while minimizing the energy usage deviation of UAV MEC servers is proposed.

### 3. System Model and Problem Formulation

In this section, the system model for the task migration method between UAV MECs in vehicular networks is defined, and the problem is formulated.

#### 3.1. Overall Architecture

Figure 1 shows the UAV MEC system proposed in this paper. We consider the environment of commuting times in urban areas where traffic temporarily increases rapidly. The system consists of ground MECs in urban areas, a set of additional UAV MECs  $M$ , and a set of vehicles  $N$ . This system only considers tasks that occur in the vehicles, and UAV MECs provide communication and computing service for vehicles on the road. In this case, it is assumed that the UAV MEC is added because the ground MEC is already overloaded. Therefore, ground MECs and UAV MECs serve the same role as edge servers, but ground MECs are not considered as targets for task migration from UAV MECs. It is assumed that all UAV MECs communicate through a wireless backhaul network and are connected to a separated federated learning server.



**Figure 1.** The architecture of the proposed UAV MEC system.

The vehicle offloads its task to the MECs with the strongest signal power. The UAV MECs migrate their tasks to the UAV MECs, which has the lowest load among its neighboring UAV MECs. However, the task will not migrate if its load status is the lowest among its neighbors. It is necessary to determine the percentage of tasks to migrate while considering the load and energy states of the current UAV MEC and the target UAV MEC. Each UAV MEC becomes an agent and trains a local model with which to determine the amount of migration tasks. The federated server collects the model parameters learned from UAV MECs and updates them to a new global model.

### 3.2. Computation, Communication, and Energy Consumption Model

During a given time slot, the vehicle  $n \in N$  offloads the task to the MECS with the strongest signal power. Each task is defined as  $\{f_n i_n, T_n^{max}\}$ .  $f_n$  is the CPU cycles required to process the task,  $i_n$  is the task size, and  $T_n^{max}$  is the delay constraint to complete the task. It is assumed that each task can be divided and migrated for execution.

The load state of the UAV MECS  $m \in M$  in the time slot  $t$  is

$$\beta_m^t = \frac{L_m^t}{L_m^{max}} \times 100 \quad (1)$$

where  $L_m^t$  is the task queue length of UAV MECS  $m$  in time slot  $t$ , and  $L_m^{max}$  is the maximum queue length of UAV MECS  $m$ .

Therefore, the load average of all MECSs in time slot  $t$  is as follows.

$$LA^t = \frac{1}{M} \cdot \sum_{m=1}^M \beta_m^t \quad (2)$$

Consequently, the load standard deviation of all MEC servers in the time slot  $t$  can be defined as

$$Load_{dev}^t = \sqrt{\frac{1}{M} \cdot \sum_{m=1}^M (|\beta_m^t - LA^t|)^2} \quad (3)$$

The data transmission rate for the task of vehicle  $n$  that is offloaded from the vehicle to the MECS  $m$  in the time slot  $t$  can be calculated as

$$R_{n,m}(t) = B_{n,m} \log_2 \left( 1 + \frac{P \cdot H_{n,m}}{\sigma^2} \right) \quad (4)$$

where  $B_{n,m}$  represents the channel bandwidth between vehicle  $n$  and UAV MECS  $m$ ,  $\sigma^2$  is the noise power between the vehicle and UAV groups,  $P$  is the transmission power of the vehicle, and  $H_{n,m}$  is the channel gain between vehicle  $n$  and UAV MECS  $m$  [14]. Therefore, the transmission delay for offloading the task from the vehicle  $n$  to UAV MECS  $m$  in time slot  $t$  is

$$T_n^{trans}(t) = \frac{i_n}{R_{n,m}(t)} \quad (5)$$

When the task is offloaded from vehicle  $n$  to UAV MECS  $m$ , it is assumed that the task is partitioned into  $k$  subtasks. Among the available subtasks, the number of subtasks to be performed on the UAV MECS  $m$  is denoted as  $k^m$ . The time required for the UAV MECS  $m$  to compute the task is defined as

$$T_{n,m}^{com}(t) = \frac{f_n \cdot \frac{k^m}{k}}{F_{n,m}} \quad (6)$$

where  $F_{n,m}$  is the computing resources allocated to vehicle  $n$  by UAV MEC  $m$  [14].

When partitioning a task offloaded from vehicle  $n$  to UAV MECS  $m$ , the number of partitioned tasks to be migrated from UAV MECS  $m$  to UAV MECS  $m'$  is referred to as  $k_{mig}^{m,m'}$ . The time required to migrate the tasks from UAV MECS  $m$  to UAV MECS  $m'$  is as follows.

$$T_{m,m'}^{mig}(t) = \frac{i_n \cdot \frac{k_{mig}^{m,m'}}{k}}{R_{m,m'}} \quad (7)$$

Next, we introduce the energy consumption model of the system. Three aspects of energy consumption for a UAV are considered: hovering energy, energy required for computation, and communication energy required for migration.



The energy required for the hovering of the UAV can be calculated as

$$E_{fly}^{t,m} = \frac{\delta \cdot \tau}{2} \cdot \|H_m^t\|^2 \quad (8)$$

where  $\delta$  is the acceleration of gravity,  $\tau$  is the length of each time slot, and  $H_m^t$  is the flight height of UAV MECS  $m$  in time slot  $t$  [14].

The energy required when the UAV  $m$  computes the offloaded task from vehicle  $n$  can be expressed as follows [14].

$$E_{com}^{t,m} = l \cdot \left( \frac{k^m}{k} \cdot F_{n,m} \right) \cdot T_{n,m}^{com}(t) \quad (9)$$

where  $l$  represents the effective switching capacitance of the CPU in a UAV MECS.

Finally, when a UAV MECS  $m$  migrates tasks to another UAV MECS  $m'$ , the required communication energy is mainly determined by the transmission power of the UAV, and it can be calculated as

$$E_{mig}^{t,m,m'} = P \cdot T_{m,m'}^{mig}(t) \quad (10)$$

Therefore, the total energy consumption of UAV  $m$  used in time slot  $t$  is as follows.

$$E_{total}^{t,m} = \varphi_1 \cdot E_{fly}^{t,m} + \varphi_2 \cdot E_{com}^{t,m} + \varphi_3 \cdot E_{mig}^{t,m,m'} \quad (11)$$

where  $\varphi_1, \varphi_2$ , and  $\varphi_3$  are weight factors ( $\varphi_1 + \varphi_2 + \varphi_3 = 1$ ).

Therefore, the energy deviation of all UAV MECSs is shown in Equation (12).  $E_{avr}^t$  means the average energy usage of UAV MECSs in time slot  $t$ .

$$E_{dev}^t = \frac{1}{M} \cdot \left( \left| E_{total}^{t,m} - E_{avr}^t \right| \right)^2 \quad (12)$$

### 3.3. Problem Formulation

In the previous subsections, a UAV MEC system for a vehicular network defined in this paper was introduced and mathematical models were established for load status, computation, communication, and energy consumption. In this subsection, the problem is formulated to achieve our goals. The three optimization goals are as follows:

First, the load deviation in UAV MECSs should be minimized. In the case of load imbalance in MECSs, the load of certain MECSs temporarily increase rapidly, and the delay constraint of tasks may not be satisfied. Therefore, load balancing allows for tasks to be completed within latency constraints.

Secondly, the energy usage deviation of UAV MECSs should be reduced. This allows the UAV MECSs to evenly use energy and avoid service area restrictions caused by the discharge of a particular MECS.

Finally, the system throughput should be maximized. System throughput refers to the number of tasks completed within the delay constraint time. In the vehicular environment, it is essential to complete the task within the delay constraint time due to the high mobility of users [18]. This is because, in a high mobility environment, the environment changes very quickly. If the vehicle moves too far while the task is being performed on the MEC server, the communication and migration cost increases, and the results cannot be obtained if the vehicle is completely out of service range. Therefore, task migration considering energy state and load state can increase task throughput and provide high quality service.

According to these goals, the objective function is defined as:

$$\max_{\alpha \cdot Load_{dev}^{t,nor} + (1-\alpha) \cdot E_{dev}^{t,nor}} Throughput \quad (13)$$

where  $Load_{dev}^{t,nor}$  and  $E_{dev}^{t,nor}$  represent values obtained by normalizing  $Load_{dev}^t$  and  $E_{dev}^t$  to values between 0 and 1. Further,  $\alpha$  is a weight factor ( $0 < \alpha < 1$ ).

#### 4. Federated DQN-Based Task Migration Algorithm

This section proposes a federated DQN-based task migration technique. This technique is intended to satisfy delay constraints and optimize throughput by determining the number of migration tasks while considering the load states and energy levels of UAV MECSs.

##### 4.1. Markov Decision Process(MDP) Formulation

In this paper, the task migration problem between UAV MECSs is formulated into MDP. Each UAV MECS becomes an agent, and each agent observes its own state and the state of the target UAV MECS. The target UAV MECS  $m'$  is assumed to have the lowest load among servers whose service coverage overlaps with the current UAV MECS. The action  $a$  is selected based on the observed environmental state  $s_t$  at time slot  $t$ , and the agent obtains the reward  $r$  for the action and then moves to the next state  $s_{t+1}$ . In this case, the state set is expressed as  $S$ , the action set is expressed as  $A$ , and the reward is expressed as  $R$ . These three MDP components are defined as follows:

- **State:** Each UAV agent observes its own state and the state of the target UAV MEC. It is assumed that each UAV MECS periodically transmits load state and energy information to neighboring UAV MECSs that share the part of service coverage. This allows the agent to optimize the policy that determines the number of migration subtasks. This policy also needs to satisfy the delay constraint of the task. Therefore, the task size and delay constraint time should be reflected, as should the load and energy states of themselves and the target UAV MECSs. Further, the upload channel state changes depending on the environment, which affects when the task is migrated to the target UAV MECS. Therefore, the SINR with the target UAV MECS is also reflected. Accordingly, the state of UAV MECS  $m$  is defined as follows.

$$s_m^t = \{i_n, T_n^{max}, \beta_m^t, E_m^t, \beta_{m'}^t, E_{m'}^t, \gamma_{m,m'}^t\} \quad (14)$$

where  $E_m^t$  and  $E_{m'}^t$  represent the residual energy of UAV MECS  $m$  and target UAV MECS  $m'$ .  $\gamma_{m,m'}^t$  is signal-to-interference-plus-noise ratio (SINR) between UAV MECS  $m$  and  $m'$ .

- **Action:** Each UAV agent allocates the number of subtasks to perform in the current MECS and the number of subtasks to migrate among the subtasks partitioned into  $k$  based on the current state. Thus, the action of UAV MEC  $m$  is defined as follows.

$$a_m^t = \{k^m, k_{mig}^{m,m'}\} \quad (15)$$

where  $k^m + k_{mig}^{m,m'} = k$ .

- **Reward:** A reward function is defined to reduce the load and energy consumption deviation among UAV MECSs and maximize the total system throughput as shown in Equation (13). Accordingly, the reward function is defined as follows.

$$r_m^t = \log_{\alpha \cdot Load_{dev}^{t,nor} + (1-\alpha) \cdot E_{dev}^{t,nor}} Throughput \quad (16)$$

Learning of the DQN model is performed in the direction of maximizing the reward value calculated through Equation (16). This reward function is a logarithmic function, and the reward value increases as the sum of load and energy deviation decreases and as the throughput increases. Therefore, according to Equation (16), the DQN model is learned in a direction of reducing load deviation and energy deviation along with increasing throughput to maximize the reward. Further, the log function increases more slowly than the linear function, and the increase in the function value slows as the input value increases. Therefore, in an environment where temporary rapid changes in vehicle mobility are likely to occur, such as in the high-mobility vehicle environment assumed in this paper, the log

function is used to prevent the task migration strategy from reacting too sensitively to temporary changes in the environment. In this way, the agent can provide further improved service quality by selecting the optimal migration strategy that maximizes the reward.

#### 4.2. DQN Algorithm

DQN, a type of reinforcement learning, is a method that uses a neural network instead of a Q-table and trains the neural network model to approximate the Q value. DQN uses a framework that receives states as inputs, exports values as outputs, and optimizes the action-value function to approximate the ideal action-value function [19]. The action-value function  $Q^*$  corresponding to the optimal policy is learned by minimizing the loss, and the loss can be expressed as follows:

$$L(\theta) = \mathbb{E}_{s,a,r,s'} \left[ \left( Q^*(s,a|\theta) - y^2 \right) \right] \quad (17)$$

$$y = r + \delta \max_{a'} Q^*(s', a') \quad (18)$$

where  $y$  is the target Q function in which the parameters are periodically updated to the most recent  $\theta$  [20], which can help stabilize learning. Another critical factor affecting DQN stabilization is the use of an experience replay buffer. The experience replay buffer stores the values for the state, action, and Q-value given at each step in the learning process of DQN, i.e., experience, and the buffer is also used to learn the network using the stored experience. With experience replay, the values listed above remain in the replay buffer and are used to update the network's weight based on these values. In this process, most of the experience data that has undergone random sampling as much as mini-batch size is extracted, and weights are updated, such that independence can be secured. Moreover, during the weight update process, the target value was changed using the Q-learning method, and weight updates are possible with new independent data for this changed value.

In the system proposed in this paper, the environment is continuously changing due to movement of the vehicle. In such an environment, it is necessary to continuously recognize changes in the environment and choose a task migration strategy that suits the situation. Further, since it is highly dynamic over time, many state factors must be reflected when determining the number of subtasks to be migrated. However, if the state space becomes too large, learning takes a substantial amount of time, thus reducing learning efficiency and performance. To solve this problem, deep reinforcement learning (DRL) can be used to recognize and respond to the highly dynamic environment in real time. DRL can predict the reward of each action using a trained neural network, thereby enabling efficient learning, even in environments with large state space dimensions [21]. Further, actions in reinforcement learning can be divided into discrete and continuous types, and DQN is known to be suitable for an environment composed of discrete actions. In this paper, the defined environment is very dynamic due to the movement of the vehicle, and the task occurrence is not regular, so it is necessary to select a migration strategy through interaction with the environment. Moreover, according to the MDP defined above, the state space is large and the action is discrete, so the DQN model is a very suitable algorithm for this environment. Therefore, in this paper, a DQN-based task migration strategy is proposed. Each UAV agent trains a neural network in a way that predicts cumulative weighted rewards for all actions. This allows the agent to choose actions that maximize rewards.

#### 4.3. Federated DQN Algorithm

In this paper, a federated DQN scheme using federated-learning-based DQN is proposed. Federated learning is a technology in which multiple local clients and a central server cooperate to learn a global model in a decentralized data environment. Federated learning has two very useful advantages: improved data privacy and communication efficiency. In situations where it is paramount to protect personal information, this enables learning without data leakage, and the communication costs are significantly reduced because only updated information of the local model is exchanged.



In addition, in the case of the system environment proposed in this paper, the model can be periodically updated for highly dynamic environments. Further, learning models can be learned by grouping regions with common geographical characteristics, and it is possible to provide effective edge services to users. It is assumed that each UAV MECS becomes an agent and that the federated learning server is separate. Each UAV MECS agent learns using a local model DQN and transmits the trained model to the federated learning server. The federated learning server aggregates these local models to create a new global optimization model. The UAV MECS agents download this model and select actions using this new model.

## 5. Simulation Results

The performance of the proposed algorithm is verified through comparative experiments with other algorithms. The simulation was implemented using Pytorch 1.11.0 with Python 3.9.7, and the vehicles were implemented using the actual vehicle mobility data, a dataset of mobility traces in San Francisco, USA [22]. To verify whether the proposed algorithm shows good performance regardless of the change in region, it is necessary to compare results that were obtained using several datasets in different regions. Therefore, experiments are conducted using additional vehicle mobility data, a dataset of mobility trace in Rome, Italy [23] and a dataset of mobility trace in Seoul, South Korea [24]. It is assumed that the experimental space is  $2.5 \text{ km} \times 2.5 \text{ km}$  and that nine UAV MECSs are placed in this space. It is considered an urban environment with a high density of vehicles. The UAV MECSs are set to hover at an altitude that can provide the widest service coverage in this environment. Therefore, UAV MECSs are assumed to fly at an altitude of 1000 m and provide a service range with a radius of 500 m [25,26]. The data size of each task is selected randomly between 2 and 6 Mbits, and the delay constraint of each task is set between 0.5 s and 2 s [27,28]. The length of one time slot is set to 60 s considering traffic congestion during commuting time. The parameters used in the experiments are summarized in Table 1 [27–29].

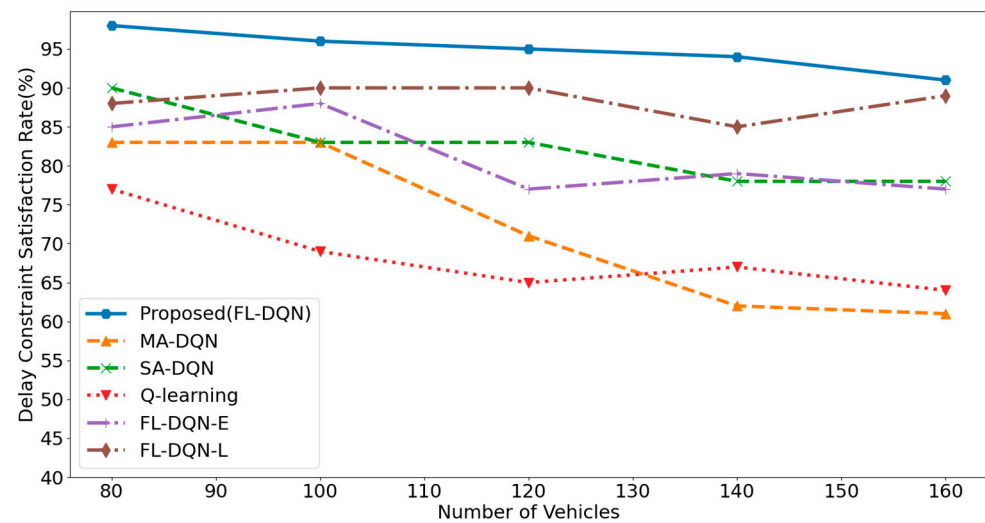
**Table 1.** Experiment parameter settings.

Parameter	Value
Number of UAV MECSs	9
Channel bandwidth ( $B$ )	1 MHz
Background noise $\sigma^2$	$10^{-9} \text{ W}$
Input data size	2–6 Mbits
Delay constraints	0.5–2 s
Maximum transmission power of vehicle $n$	2 W
Length of time slot ( $\tau$ )	60 s

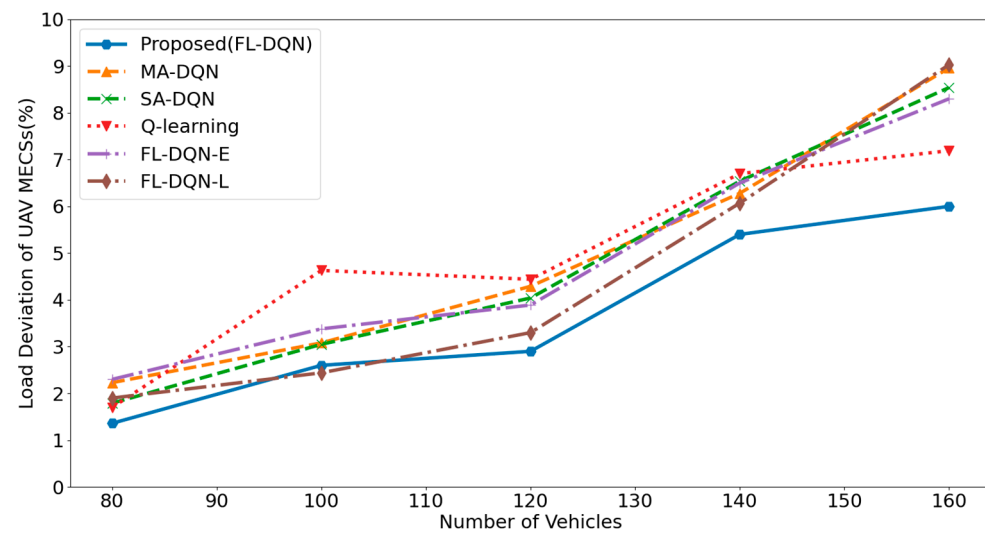
To evaluate the performance of the proposed federated DQN (FL-DQN) algorithm, it is compared with multi-agent DQN (MA-DQN)- and single-agent DQN (SA-DQN)-based migration methods. To evaluate the performance of DQN that uses a neural network instead of a Q-table, it is compared with a Q-learning-based task migration method [30]. In addition, the proposed technique is compared with two scenarios to confirm whether it is adequate to consider both load deviation and energy deviation. Only load deviation is considered in one case (FL-DQN-L), and only energy deviation is considered in another case (FL-DQN-E).

Figures 2–4 show the measured results when changing the number of vehicles in the experimental environment. In this case, the vehicle's speed is fixed at  $5 \frac{\text{m}}{\text{s}}$ , and the number of task partitions is fixed at 10. As the number of vehicles increases, the number of tasks that occur increases. Figure 2 shows the satisfaction rate of delay constraints with a varying number of vehicles in the environment. The proposed algorithm has the highest delay constraint satisfaction, and it performs approximately 21–22% better than MA-DQN and approximately 11–12% better than SA-DQN. This is because in the multi-agent method each

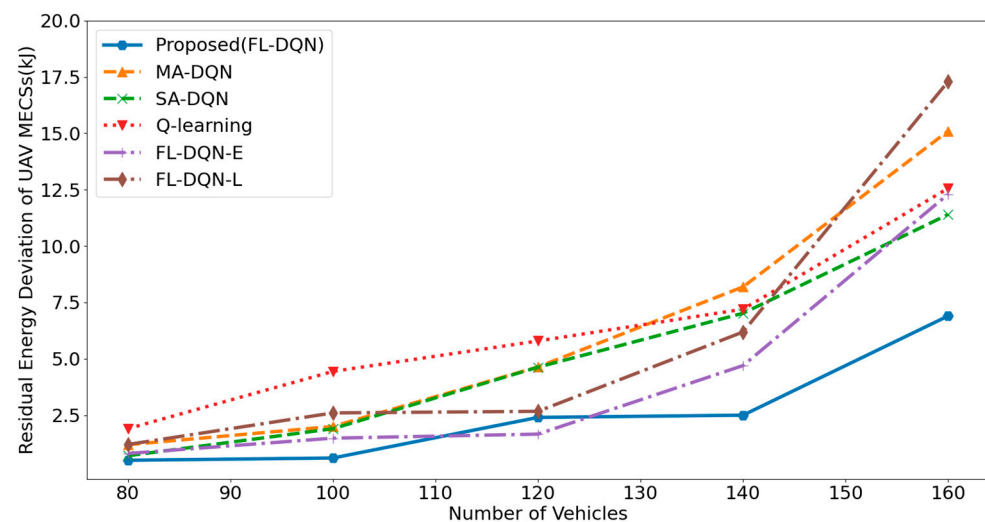
UAV MECS learns its model through observed states and does not share model weights with other UAV MECSs. In the case of SA-DQN, since the central agent has to collect the state information of all UAV MECSs and decide the action for each one, the network traffic and storage costs increase, which makes it challenging to determine the optimal solution. Further, DQN-based algorithms show higher performance than Q-learning-based algorithms. This is because DQN enables efficient learning by solving the problem of the table size of Q-learning becoming too enormous and the correlation problem between sample data of Q-learning. While the number of tasks increases as the number of vehicles increases, the number of tasks that UAV MECS can handle is limited. Therefore, as the number of vehicles increases, it becomes difficult for the UAV MEC to handle all tasks within delay constraints. This leads to a decrease in the satisfaction of the delay constraint, which results in the overall decrease in the satisfaction of the delay constraint shown in Figure 2. At this time, the delay constraint satisfaction of the proposed algorithm decreases in a relatively linear manner as the number of vehicles increases, unlike other techniques, and changes to the smallest width. The reason for this is that the proposed method enables faster and more efficient learning using neural networks instead of a Q-table, and it creates a global model suitable for the entire environment through federated learning. In the case of Q-learning using the Q-table, as the Q-table grows larger due to the large state space, memory problems occur, and learning takes too much time, which makes learning difficult in practice. DQN uses a neural network, and it trains the neural network model to approximate the Q value. This means that efficient learning is possible even in an environment with a large state space, and stable performance can be achieved even with an increased number of vehicles. However, other techniques may degrade system performance in the process of training the model. For example, in a multi-agent environment, each agent creates a model suitable for each service range, but the throughput of the entire service range may decrease because the situation of different agents cannot be known. Since the task is migrated in consideration of only its own state, the processing delay further increases as the number of vehicles increases and the number of tasks increases. Therefore, as shown in Figure 2, the performance of MA-DQN deteriorates most rapidly. In addition, in the case of a single agent, it is necessary to select an appropriate migration strategy by collecting information on all UAV MECS existing within the entire service coverage. This requires too much information and consumes a lot of time. FL\_DQN\_L and FL\_DQN\_E also use federated learning. However, the reason why the change is not linear and fluctuates is because when only one of the load and energy variation is considered, the result is not constant, depending on the environment. In the case of FL\_DQN\_L, which only considers the load state, it consumes a lot of energy to reduce the load deviation, thus resulting in an energy exhaustion problem and consequently affecting throughput. In addition, in the case of FL\_DQN\_E, processing delay occurs because load balancing is not well-performed when considering only energy. Therefore, stable service cannot be provided according to changes in the environment, and there are variations depending on various factors, such as the location of the vehicle. In this way, it can be confirmed that the proposed technique can provide efficient and fast service in any environment. Figures 3 and 4 show the load and residual energy deviation between UAV MECSs according to the number of vehicles. Comparing FL-DQN with FL-DQN-L and FL-DQN-E can confirm that it can provide the most effective service considering both load deviation and energy deviation. This is because energy is not evenly used when only the load deviation is considered, which results in defects within the entire service provision range, and load imbalance occurs when only the energy deviation is considered. As the proposed algorithm's load deviation and energy deviation are very low, the load balancing in consideration of energy usage deviation is performed well compared to other techniques. However, as the number of vehicles increases, the delay constraint satisfaction decreases, and the load deviation and energy deviation increase. This is because the number of tasks to be processed increases as the number of vehicles increases, but the computing power of the UAV MECS is limited.



**Figure 2.** Delay constraint satisfaction rate with varying number of vehicles.

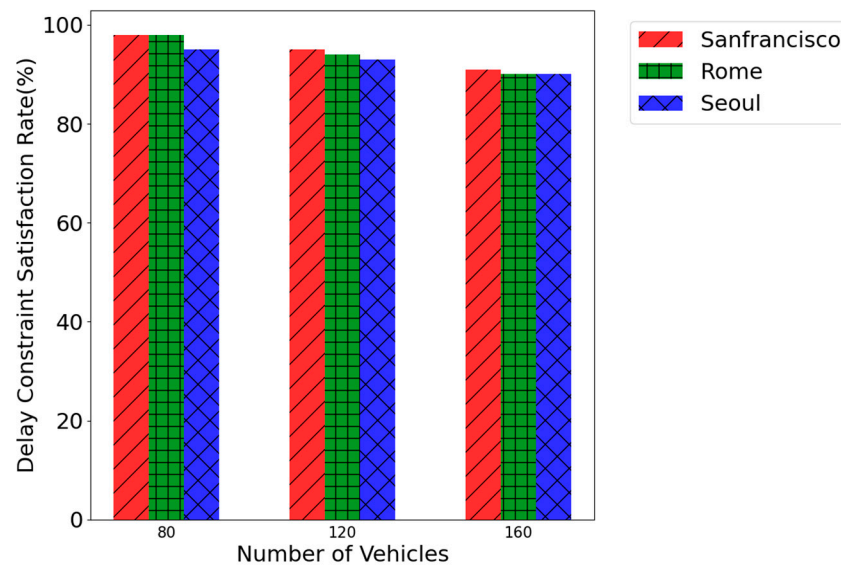


**Figure 3.** Load deviation of UAV MECs with varying number of vehicles.

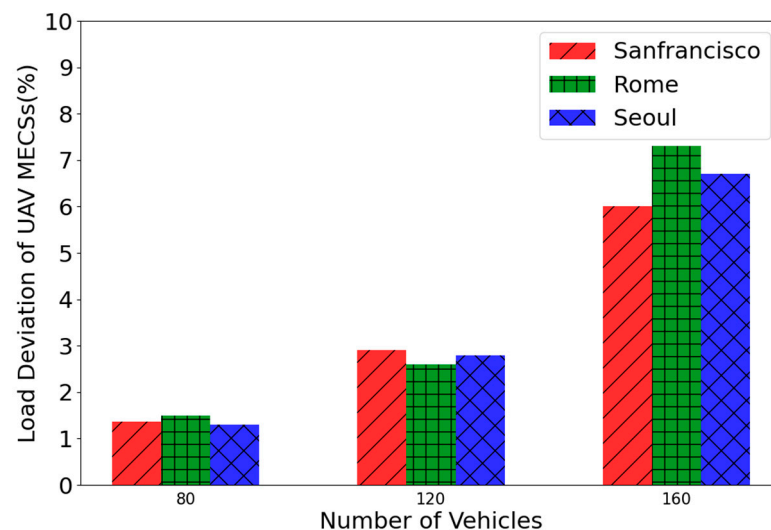


**Figure 4.** Residual energy deviation of UAV MECs with varying number of vehicles.

Figures 5–7 show the measurement results when changing the number of vehicles in different environments. According to Figure 5, even if the experimental environment changes, the proposed algorithm shows a delay constraint satisfaction rate of about 94–98%. Figures 6 and 7 show that the load and energy differences among the UAV MECs are both very small. Since the proposed algorithm selects an appropriate migration strategy by reflecting changes in the environment, it shows similar performance even when using other datasets. In this way, it can be confirmed that the proposed algorithm shows good performance in various regions or environments where the number and location of vehicles are different.



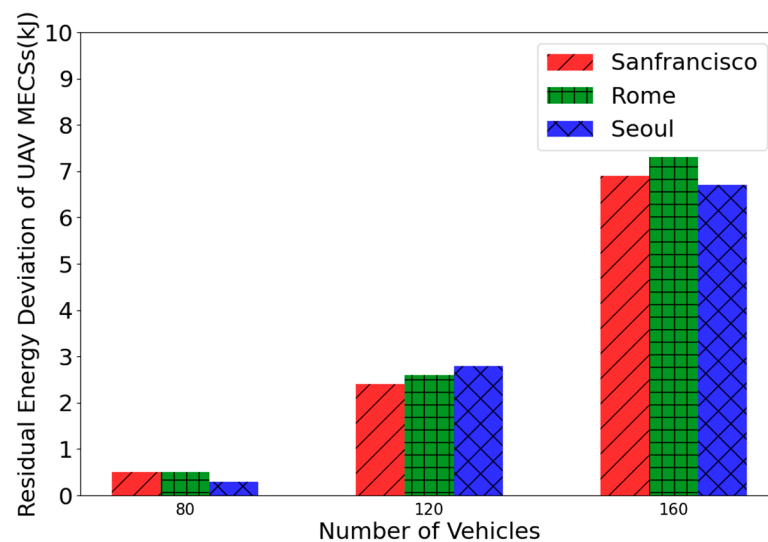
**Figure 5.** Delay constraint satisfaction rate with varying number of vehicles in different environments.



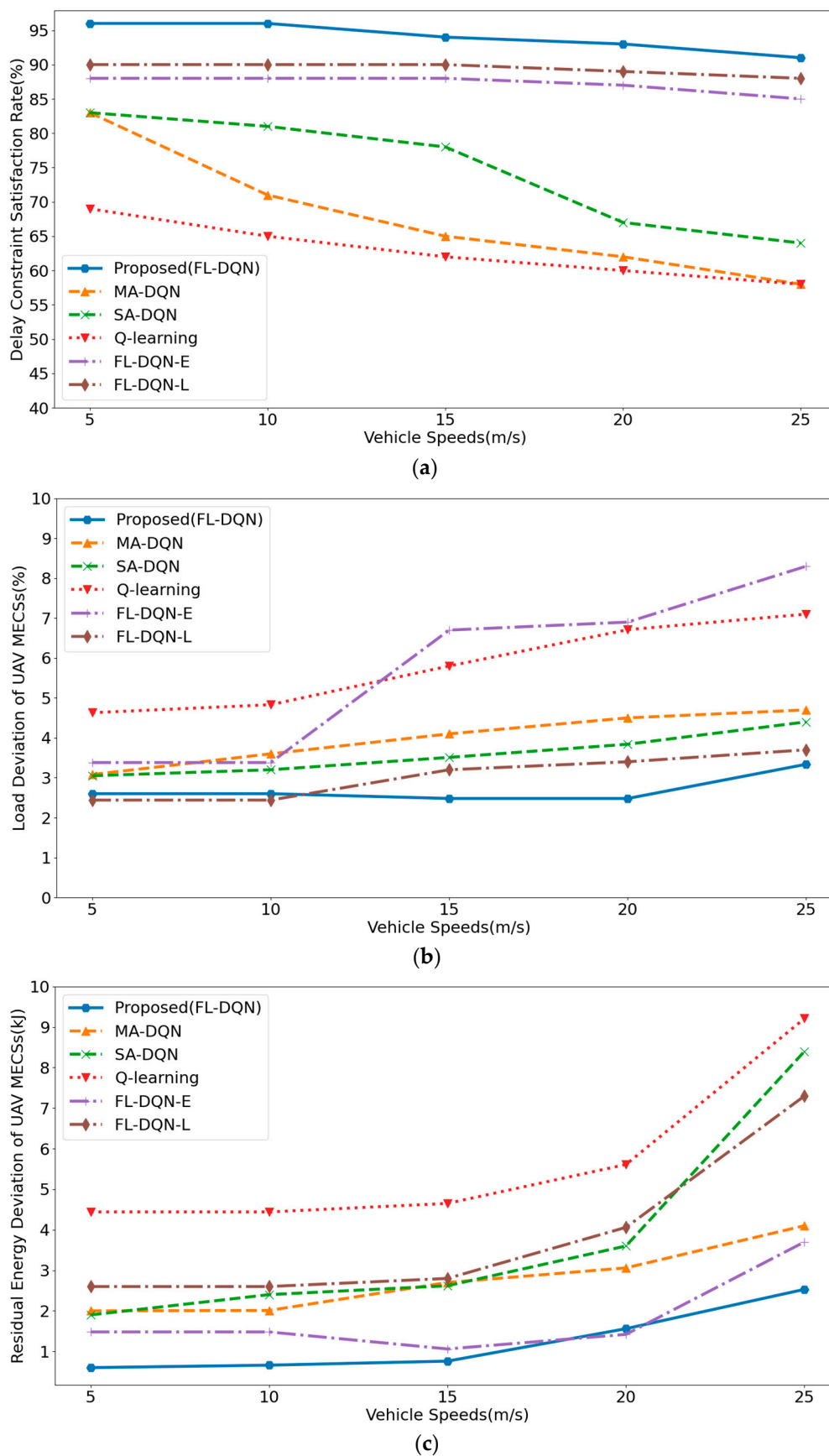
**Figure 6.** Load deviation of UAV MECs with varying number of vehicles in different environments.

Figure 8 shows the measured results when the number of vehicles used in the experimental environment is set to 100, the number of task partitions is fixed to 10, and the movement speed of the vehicles is varied. As the speed of the vehicle increases, the environment becomes more dynamic. Figure 8a shows the satisfaction ratio of delay constraints according to the average vehicle speed. Delay constraint satisfaction is a number that is directly related to overall system throughput. Compared to other algorithms, the proposed algorithm has the highest satisfaction in delay constraints and shows about 19–20% better performance than SA-DQN and about 26–27% better performance than MA-DQN. It

provides relatively stable service, as there is no significant difference in the satisfaction of delay constraints even as the speed of the vehicle increases. This demonstrates that the proposed algorithm periodically updates the model in dynamically changing environments, so even if the vehicle moves fast, tasks can be migrated through appropriate learning. This is because the proposed algorithm is based on FL, meaning that unlike DQN techniques that only learn the local optical model, it learns the global optical model. In addition, SA-DQN performs about 11–12% better than Q-learning, indicating that DQN is more suitable than Q-learning in environments where the user movement is dynamic and the state space is enormous. Figure 8b shows the load deviation between UAV MECs according to vehicle speed. The proposed algorithm shows the smallest load deviation, which means the algorithm balances loads of UAV MECs well. Since FL\_DQN\_L only considers the load deviation without considering the energy deviation, the load deviation may be smaller than FL\_DQN in certain circumstances. In Figure 8b, when the speed of the vehicle is slow, the load deviation of FL\_DQN\_L is smaller than the proposed load deviation of FL\_DQN for this reason. However, the energy is not used evenly, and it causes rapid energy consumption when migrating tasks to reduce only load variation. In particular, as the vehicle speed increases, it becomes more difficult to satisfy the delay constraint, thus resulting in more task migrations without considering energy to reduce load deviation while increasing throughput. As can be seen in Figure 8c, the residual energy deviation increases significantly as the vehicle speed increases. This means that only certain UAV MECs are consuming more energy to reduce the load deviation as the speed of the vehicle increases. At this time, if the energy of a specific UAV MEC is exhausted, the corresponding MEC can no longer provide service, and the load deviation becomes larger. It also affects the overall system throughput. Therefore, even if there is a situation where the load deviation of FL\_DQN\_L is smaller, it is necessary to consider both load deviation and energy deviation to provide more continuous and stable service in a dynamic environment. Figure 8c shows the residual energy deviation between UAV MECs with varying vehicle speeds. In terms of energy deviation and load deviation, the proposed algorithm shows the lowest deviations and thus ensures that the load is balanced while using energy more equally than other algorithms.



**Figure 7.** Residual energy deviation of UAV MECs with varying number of vehicles in different environments.



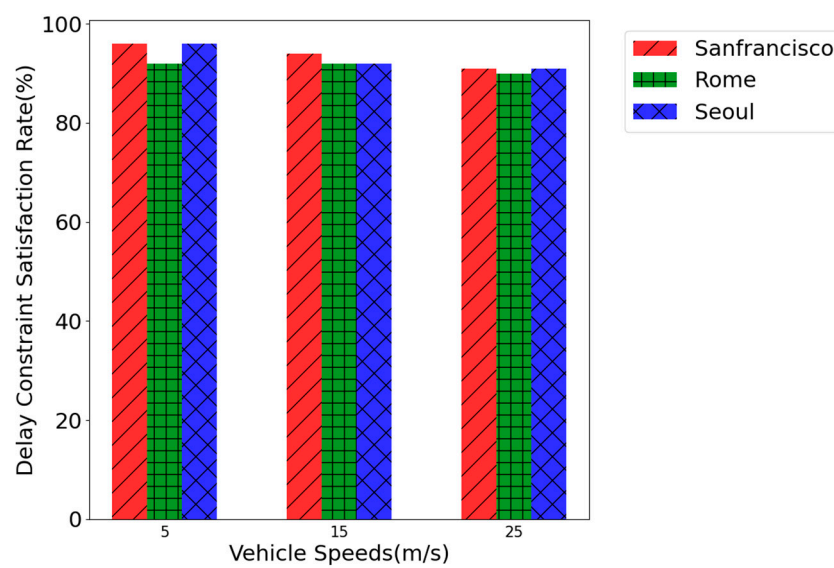
**Figure 8.** Experimental results with varying vehicle speeds: (a) delay constraint satisfaction rate; (b) load deviation of UAV MECs; (c) residual energy deviation of UAV MECs.



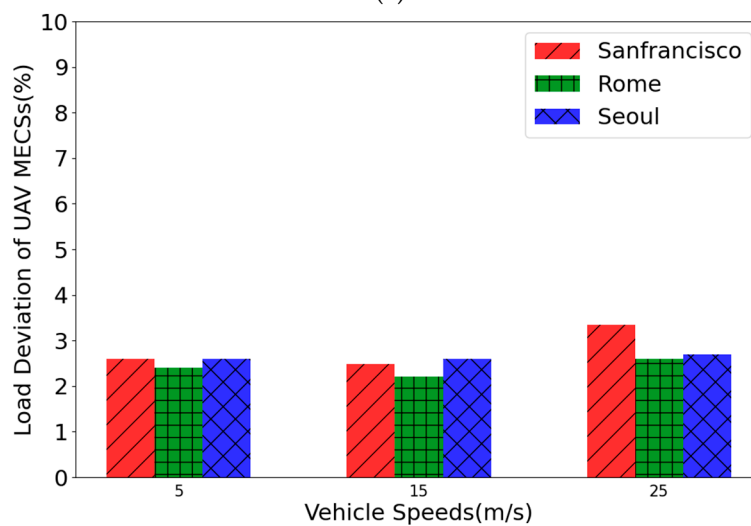
Figure 9 shows the measured results when the mobility speed of the vehicles is varied in different environments. As shown in Figure 9a, the proposed algorithm has high delay constraint satisfaction rates of about 94–98% even when the vehicle speed changes regardless of the type of dataset. According to Figure 9b,c, the load and energy deviations among UAV MECSs in all data sets also do not show significant differences. It can therefore be confirmed that the proposed technique can provide an effective service by learning a model suitable for the situation even if the characteristics of the environment or the mobility of the vehicle are different.

Figure 10 shows the measured results as the number of subtasks divided by the task changes. In this case, the number of vehicles is 100, and the average speed of the vehicle is 5 m/s. As shown in Figure 10a, as the number of subtasks to be partitioned changes, the proposed algorithm shows the highest satisfaction with delay constraints. However, in most technologies, as the number of subtasks increase, the delay constraint satisfaction rate increases and then decreases again. As the number of subtasks increases, the subtask size decreases, but the amount of data moving through the network increases, thus increasing overall network traffic. Therefore, it is only good in some cases to have the largest or smallest number of subtasks, but the optimal number of subtasks suitable for each environment varies. Figure 10b,c show that the proposed algorithm distributed the load evenly among the UAV MECSs and used the energy equally even though the number of subtasks changed compared to other techniques. This is because the proposed algorithm can learn more efficiently than Q-learning in most situations, even if the number of subtasks changes; unlike other algorithms, it learns the optimal global model through federated learning. As shown in Figure 10b, the proposed algorithm has higher load deviation for the lower number of sub-tasks compared to FL-DQN-L. As mentioned above, the load deviation of FL\_DQN\_L considering only the load deviation may be smaller than the load deviation of FL\_DQN in some circumstances. This is only possible in a situation where all UAV MECS provide services without energy consumption, even considering only load deviation. However, in most situations, the energy depletion of a particular UAV MECS will occur if it migrates tasks without considering the energy deviation. The energy depletion of certain UAV MECS results in an increase in load deviation. As the number of subtasks increases, overall network traffic increases, thus resulting in more frequent migrations. Therefore, a UAV MECS that exhausts all energy may occur. As such, the load state and the energy state directly affect each other, and it is necessary to consider both the energy state and the load state to ensure that all UAV MECS provide services continuously and evenly.

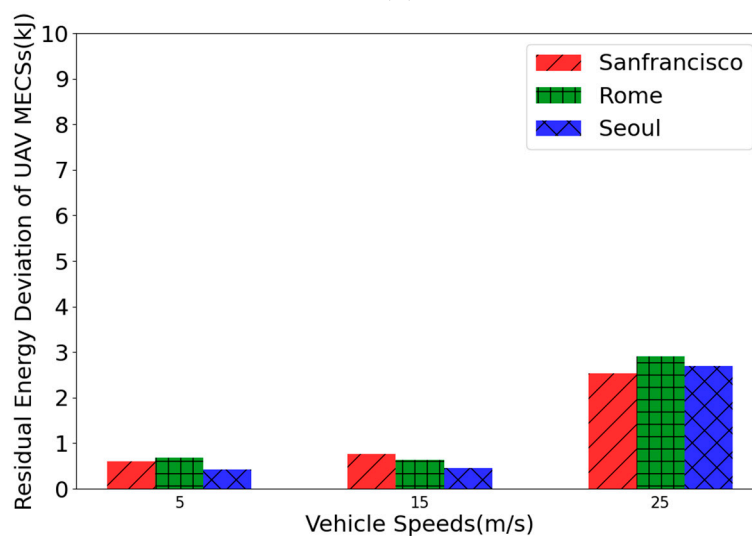
Finally, an experiment was conducted to verify whether the federated DQN-based collaborative migration strategy proposed in this paper is realistic for practical application. The purpose of this paper is to increase the throughput by distributing the load and to use the energy of the UAV MEC evenly to provide services without depleting the energy of all UAVs. Ultimately, it is intended to allow all UAV MECSs to provide full service during the entire time the UAV MECS is required. Therefore, it is necessary to measure the time that all UAV MECSs can provide service without energy depletion. Figure 11 shows the experimental results during the time that all UAV MECSs can provide service without energy exhaustion. The value shown in the chart is the service duration measured when the energy remaining after subtracting the energy for UAV flight from the UAV initial energy is used for task processing and communication. It is assumed that the duration of service ends when at least one UAV runs out of energy. The experimental results confirm that the service duration of the proposed algorithm is 16–33 min longer than those of other techniques. This is because the proposed algorithm selects the most appropriate migration strategy for the environment through effective model learning and updating. By reducing energy deviation, all UAV MECSs can use energy evenly. This increases the service duration of all UAV MECS by delaying the time until the energy of a specific UAV MECS is exhausted. In this way, it can be confirmed that the proposed technique uses energy evenly and can serve the longest period compared to other techniques. This means that the proposed algorithm can be used effectively in practice.



(a)

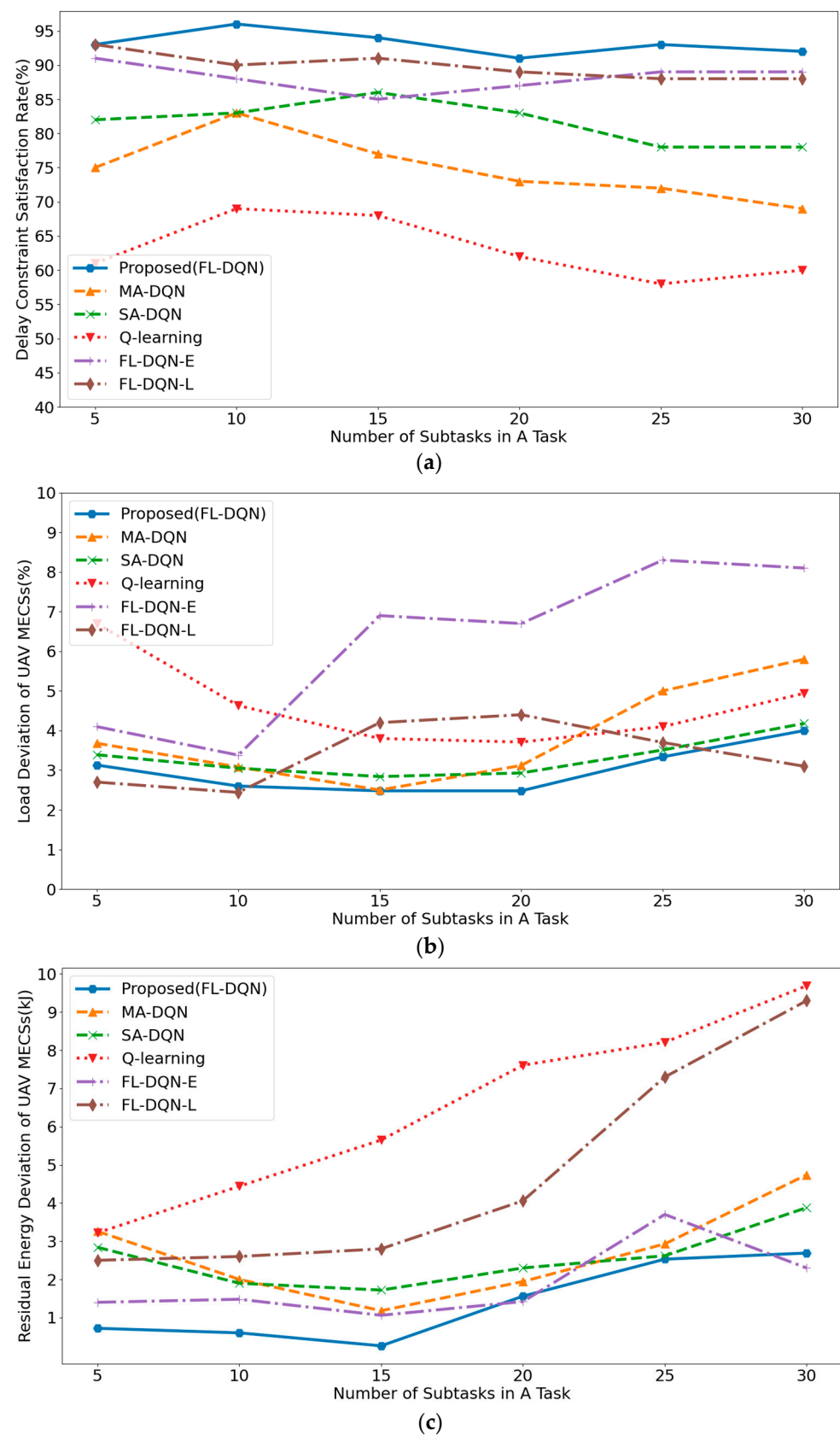


(b)

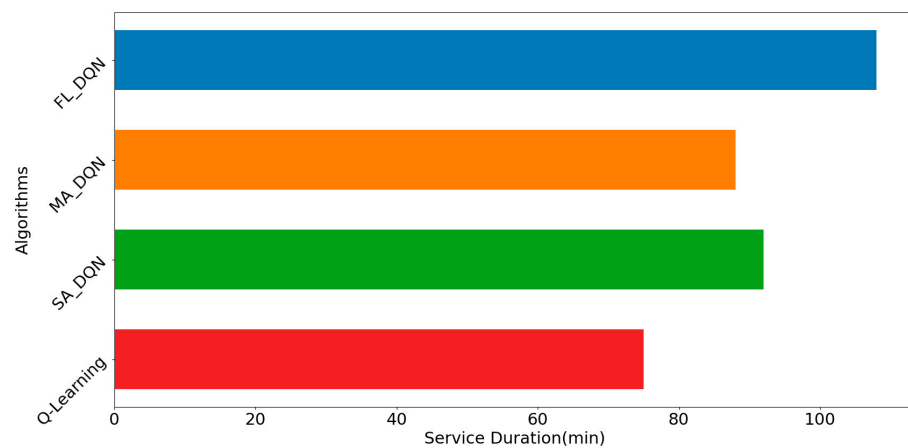


(c)

**Figure 9.** Experimental results with varying vehicle speeds in different environments: (a) delay constraint satisfaction rate; (b) load deviation of UAV MECs; (c) residual energy deviation of UAV MECs.



**Figure 10.** Experimental results with varying number of subtasks in a task: (a) delay constraint satisfaction rate; (b) load deviation of UAV MECs; (c) residual energy deviation of UAV MECs.



**Figure 11.** Service duration that can be provided without energy exhaustion.

## 6. Conclusions and Future Research Directions

This paper has proposed a method for effectively handling the dynamics of task processing requests in a vehicular environment with computationally intensive and latency-sensitive tasks. A federated DQN-based migration algorithm was proposed to minimize load and energy deviation in UAV MECs and maximize overall system throughput. DQN can be used to build a local model for migration optimization for each UAV MEC. Federated learning can build an effective global model between adjacent regions with common spatial features. Experiments considering multiple scenarios demonstrate that the proposed algorithm performs well in terms of delay constraint satisfaction, load deviation, and energy deviation. The task migration and communication costs for each UAV MEC increase as the number of vehicles increases or their speed increases. This can affect the overall system throughput and increase the energy consumption of UAV MECs. However, the proposed technique optimizes the migration strategy by considering the energy and load conditions so that the cost does not increase exponentially with environmental changes. This makes it possible to provide stable high-quality service even if the number of vehicles or speed changes, as has been verified through experiments on various situations. However, in this study, the UAVs to be migrated have been determined without considering the moving direction of the vehicle. If the vehicle that generated the task and the UAV that will process the task are moving in opposite directions, data transmission may require more energy and time. Therefore, in future studies, load balancing among the UAV MEC servers considering the movement directions of vehicles will be studied. If the work moves by predicting the movement direction of the vehicle in advance, more efficient service can be provided.

**Author Contributions:** Conceptualization, A.S. and Y.L.; methodology, A.S.; software, A.S.; writing—review and editing, A.S. and Y.L.; supervision, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) program (IITP-2022-RS-2022-00156299) supervised by the IITP (Institute of Information and Communications Technology Planning and Evaluation), and also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1047113).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- De Souza, A.B.; Rego, P.A.L.; Carneiro, T.; Rodrigues, J.D.C.; Filho, P.P.R.; De Souza, J.N.; Chamola, V.; De Albuquerque, V.H.C.; Sikdar, B. Computation Offloading for Vehicular Environments: A Survey. *IEEE Access* **2020**, *8*, 198214–198243. [\[CrossRef\]](#)
- Zhang, J.; Letaief, K.B. Mobile Edge Intelligence and Computing for the Internet of Vehicles. *Proc. IEEE* **2020**, *108*, 246–261. [\[CrossRef\]](#)
- Dai, Y.; Xu, D.; Maharjan, S.; Zhang, Y. Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2019**, *6*, 4377–4387. [\[CrossRef\]](#)
- Peng, H.; Shen, X.S. DDPG-Based Resource Management for MEC/UAV-Assisted Vehicular Networks. In Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 18 November 2020–16 December 2020; pp. 1–6. [\[CrossRef\]](#)
- Abrar, M.; Ajmal, U.; Almohaimeed, Z.M.; Gui, X.; Akram, R.; Masroor, R. Energy Efficient UAV-Enabled Mobile Edge Computing for IoT Devices: A Review. *IEEE Access* **2021**, *9*, 127779–127798. [\[CrossRef\]](#)
- He, X.; Meng, M.; Ding, S.; Li, H. A Survey of Task Migration Strategies in Mobile Edge Computing. In Proceedings of the 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 24–26 April 2021; pp. 400–405. [\[CrossRef\]](#)
- Pandey, D.; Pandey, P. Approximate Q-Learning: An Introduction. Available online: <https://ieeexplore.ieee.org/document/5460718> (accessed on 28 August 2022). [\[CrossRef\]](#)
- Yuan, Q.; Li, J.; Zhou, H.; Lin, T.; Luo, G.; Shen, X. A Joint Service Migration and Mobility Optimization Approach for Vehicular Edge Computing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9041–9052. [\[CrossRef\]](#)
- Peng, Y.; Tang, X.; Zhou, Y.; Li, J.; Qi, Y.; Liu, L.; Lin, H. Computing and Communication Cost-Aware Service Migration Enabled by Transfer Reinforcement Learning for Dynamic Vehicular Edge Computing Networks. *IEEE Trans. Mob. Comput.* **2022**, 1–12. [\[CrossRef\]](#)
- Abouamar, A.; Mlika, Z.; Filali, A.; Cherkaoui, S.; Kobbane, A. A Deep Reinforcement Learning Approach for Service Migration in MEC-Enabled Vehicular Networks. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 4–7 October 2021; pp. 273–280. [\[CrossRef\]](#)
- Wang, C.; Peng, J.; Jiang, F.; Zhang, X.; Liu, W.; Gu, X.; Huang, Z. An Adaptive Deep Q-Learning Service Migration Decision Framework for Connected Vehicles. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 944–949. [\[CrossRef\]](#)
- Peng, Y.; Liu, L.; Zhou, Y.; Shi, J.; Li, J. Deep Reinforcement Learning-Based Dynamic Service Migration in Vehicular Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [\[CrossRef\]](#)
- Du, Z.; Wu, C.; Yoshinaga, T.; Yau, K.-L.A.; Ji, Y.; Li, J. Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues. *IEEE Open J. Comput. Soc.* **2020**, *1*, 45–61. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ouyang, W.; Chen, Z.; Wu, J.; Yu, G.; Zhang, H. Dynamic Task Migration Combining Energy Efficiency and Load Balancing Optimization in Three-Tier UAV-Enabled Mobile Edge Computing System. *Electronics* **2021**, *10*, 190. [\[CrossRef\]](#)
- Gong, C.; Wei, L.; Gong, D.; Li, T.; Feng, F. Energy-Efficient Task Migration and Path Planning in UAV-Enabled Mobile Edge Computing System. *Complexity* **2022**, *2022*, 1–16. [\[CrossRef\]](#)
- Zhu, S.; Gui, L.; Cheng, N.; Zhang, Q.; Sun, F.; Lang, X. UAV-Enabled Computation Migration for Complex Missions: A Reinforcement Learning Approach. *IET Commun.* **2020**, *14*, 2472–2480. [\[CrossRef\]](#)
- Grasso, C.; Raftopoulos, R.; Schembra, G.; Serrano, S. H-HOME: A Learning Framework of Federated FANETs to Provide Edge Computing to Future Delay-Constrained IoT Systems. *Comput. Netw.* **2022**, *219*, 109449. [\[CrossRef\]](#)
- Zhang, X.; Zhang, J.; Liu, Z.; Cui, Q.; Tao, X.; Wang, S. MDP-Based Task Offloading for Vehicular Edge Computing under Certain and Uncertain Transition Probabilities. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3296–3309. [\[CrossRef\]](#)
- Morales, M. *Grokking Deep Reinforcement Learning*; Manning Publications: Shelter Island, NY, USA, 2020.
- Wang, Y.; Liu, H.; Zheng, W.; Xia, Y.; Li, Y.; Chen, P.; Guo, K.; Xie, H. Multi-Objective Workflow Scheduling with Deep-Q-Network-Based Multi-Agent Reinforcement Learning. *IEEE Access* **2019**, *7*, 39974–39982. [\[CrossRef\]](#)
- Gao, Z.; Jiao, Q.; Xiao, K.; Wang, Q.; Mo, Z.; Yang, Y. Deep Reinforcement Learning Based Service Migration Strategy for Edge Computing. In Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) 2019, San Francisco, CA, USA, 4–9 April 2019. [\[CrossRef\]](#)
- Piorkowski, M.; Sarafijanovic-Djukic, N.; Grossglauser, M. CRAWDAD Dataset Epfl/Mobility (V.2009-02-24). 2009. [\[CrossRef\]](#)
- Bracciale, L.; Bonola, M.; Loreti, P.; Bianchi, G.; Amici, R.; Rabuffi, A. CRAWDAD Dataset Roma/Taxi (v. 17 July 2014). 2014. [\[CrossRef\]](#)
- Kumbhar, F.H. *Vehicular Mobility Trace at Seoul, South Korea*; IEEE Dataport: Piscataway, NJ, USA, 2020. [\[CrossRef\]](#)
- Holis, J.; Pechac, P. Elevation Dependent Shadowing Model for Mobile Communications via High Altitude Platforms in Built-up Areas. *IEEE Trans. Antennas Propag.* **2008**, *56*, 1078–1084. [\[CrossRef\]](#)
- Al-Hourani, A.; Kandeepan, S.; Lardner, S. Optimal LAP Altitude for Maximum Coverage. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 569–572. [\[CrossRef\]](#)
- Yang, C.; Liu, Y.; Chen, X.; Zhong, W.; Xie, S. Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks. *IEEE Access* **2019**, *7*, 26652–26664. [\[CrossRef\]](#)

28. Zhang, J.; Guo, H.; Liu, J.; Zhang, Y. Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2092–2104. [[CrossRef](#)]
29. Cheng, K.; Teng, Y.; Sun, W.; Liu, A.; Wang, X. Energy-Efficient Joint Offloading and Wireless Resource Allocation Strategy in Multi-MEC Server Systems. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [[CrossRef](#)]
30. Cao, S.; Wang, Y.; Xu, C. Service Migrations in the Cloud for Mobile Accesses: A Reinforcement Learning Approach. In Proceedings of the 2017 International Conference on Networking, Architecture, and Storage (NAS), Shenzhen, China, 7–9 August 2017; pp. 1–10. [[CrossRef](#)]

**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.