Computers, Materials & Continua
DOI: 10.32604/cmc.2023.034892
Article

DQN-Based Proactive Trajectory Planning of UAVs in Multi-Access Edge Computing

Adil Khan[1,*], Jinling Zhang[1], Shabeer Ahmad[1], Saifullah Memon[2], Babar Hayat[1] and Ahsan Rafiq[3]

[1]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[3]Department of Automation, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
*Corresponding Author: Adil Khan. Email: adil@bupt.edu.cn
Received: 31 July 2022; Accepted: 22 September 2022

Abstract: The main aim of future mobile networks is to provide secure, reliable, intelligent, and seamless connectivity. It also enables mobile network operators to ensure their customer's a better quality of service (QoS). Nowadays, Unmanned Aerial Vehicles (UAVs) are a significant part of the mobile network due to their continuously growing use in various applications. For better coverage, cost-effective, and seamless service connectivity and provisioning, UAVs have emerged as the best choice for telco operators. UAVs can be used as flying base stations, edge servers, and relay nodes in mobile networks. On the other side, Multi-access Edge Computing (MEC) technology also emerged in the 5G network to provide a better quality of experience (QoE) to users with different QoS requirements. However, UAVs in a mobile network for coverage enhancement and better QoS face several challenges such as trajectory designing, path planning, optimization, QoS assurance, mobility management, etc. The efficient and proactive path planning and optimization in a highly dynamic environment containing buildings and obstacles are challenging. So, an automated Artificial Intelligence (AI) enabled QoS-aware solution is needed for trajectory planning and optimization. Therefore, this work introduces a well-designed AI and MEC-enabled architecture for a UAVs-assisted future network. It has an efficient Deep Reinforcement Learning (DRL) algorithm for real-time and proactive trajectory planning and optimization. It also fulfills QoS-aware service provisioning. A greedy-policy approach is used to maximize the long-term reward for serving more users with QoS. Simulation results reveal the superiority of the proposed DRL mechanism for energy-efficient and QoS-aware trajectory planning over the existing models.

Keywords: Multi-access edge computing; UAVs; trajectory planning; QoS assurance; reinforcement learning; deep Q network

1 Introduction

Due to the continuously increasing growth of smart devices such as smart sensors, smartphones, and wearable smart devices, many intelligent and smart applications such as gaming, artificial and virtual reality (AR-VR), and computer vision applications have emerged on edge [1]. Multi-access Edge Computing (MEC) emerged as an innovative technology for accommodating the need of constrained devices and assuring the quality of service (QoS) to the customers. MEC enhances the network edge's computing capacity and allows the deployment of cutting-edge applications and services efficiently and flexibly for large-scale smart devices [2,3]. By deploying the MEC, the smart devices can move their computationally demanding tasks to the nearest powerful edge servers, conserving energy and lowering latency. Moreover, network function virtualization (NFV) and software-defined networking (SDN) enhanced the MEC capabilities by enabling the deployment of virtual network functions (VNF) on edge [4–6], for example, pilot and control functions of UAVs. Recent research on MEC has focused on mobile edge servers rather than fixed servers because they can offer cost-effective, highly flexible, and efficient computation services in a challenging environment. Although, MEC is being developed to increase the computation capabilities of smart devices to carry out high computation and latency-critical jobs. But it still confronts other challenges, including computation improvement, energy conservation, and latency assurance. Numerous initiatives have been made to investigate these problems in MEC systems [7–10].

Several recent works have proposed using UAVs to increase connection to the ground users (GUs) and provide services to them, such as mobile and Internet of Things (IoT) devices with low computation power [11,12]. The use of UAVs in MEC gained much interest because of their advantages in improving network capacity, performance, flexible deployment, and full mobility control [13]. UAV-enabled MEC has numerous unique features that set it apart from the terrestrial servers. First, UAVs can change their locations according to the real-time offloading strategies of users. Its trajectory can be precisely planned for various objectives, including throughput improvement and energy conservation. Additionally, UAVs often avoid the geography effect due to their high altitude, strengthening and enhancing the coverage. Due to the high probability of line of sight (LoS) linkages with GUs, UAVs are less impacted by channel limitations. With the help of these features, UAVs can contribute significantly to MEC systems and overcomes terrestrial server deployment deficiencies [14–17]. UAV-enabled edge computing is an obvious and viable option for future networks. Fig. 1 illustrates the advantage of UAVs-enabled MEC over the fixed edge servers. UAVs-enabled edge provides seamless service connectivity with low latency.

Deep Reinforcement Learning (DRL) is a recent advancement in the Machine Learning (ML) field, which is a combination of Deep Neural Networks (DNNs) and Reinforcement Learning (RL). In DRL, an agent interacts with the unknown environment to find the best policy through exploration. By utilizing the strength of DNNs, DRL enables more stable approximation and convergence for calculating the associated functions compared to traditional RL approaches [18–21]. DRL is widely adopted in the network research area for solving complex problems such as resource management, energy-saving, power allocation, efficient routing, traffic management, etc. More specifically, from the recent literature [8,17] on UAVs-enabled MEC, DRL has been used for trajectory planning, energy management, user scheduling, resource management, resource provisioning, bit allocation, and throughput maximization. However, the efficient and optimal path planning in UAVs to accommodate the processing and communication in a large variety of devices is still a crucial and challenging problem.

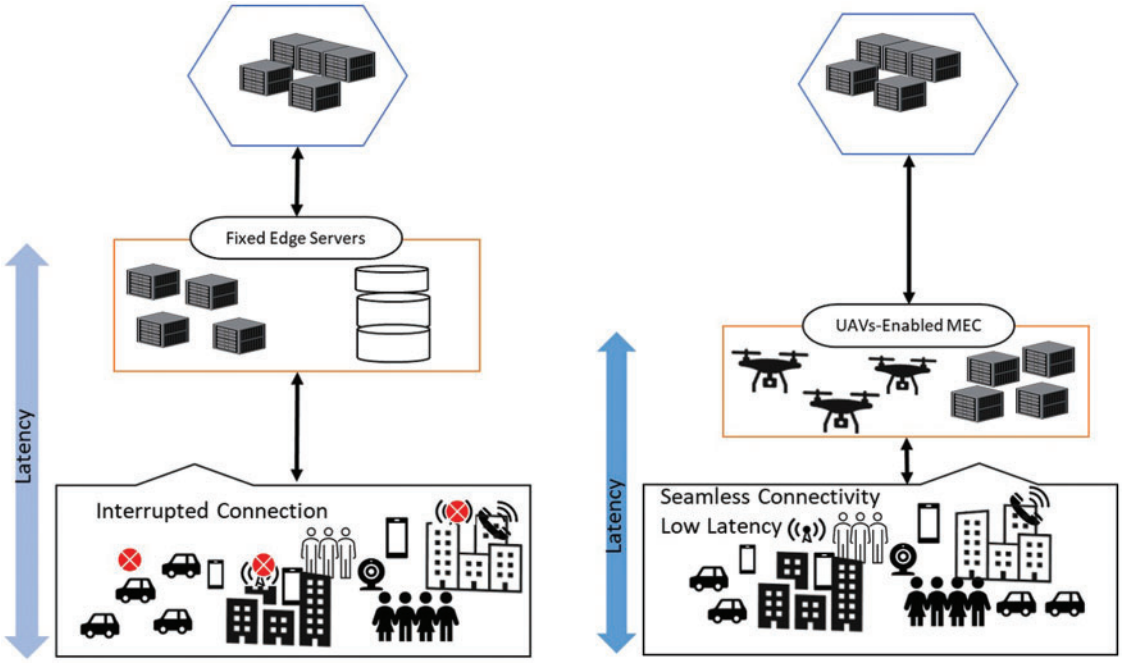

Figure 1: Comparison of UAVs-enabled MEC with a fixed edge computing environment

This paper introduces intelligent MEC and UAVs-enabled network architecture, which provides optimized service on edge and cloud. It contains Artificial intelligence (AI) enabled network service and analytics module, which includes multiple AI algorithms for efficient service provisioning to the end-users. The use of AI technologies in the network ensures proactive management of network resources and fulfills service level agreement (SLA). The AI algorithms can predict user behavior, traffic volume, efficient resource usage prediction, UAVs path planning, path optimization, mobility management, and many more. It provides intelligent services to UAVs, including trajectory planning, mobility management, and placement decision. Moreover, MEC provides extreme edge capabilities to the GUs, that can be achieved cost-effectively through UAVs. As flying MEC servers, UAVs are the best option in disaster or emergency scenarios or to enhance network coverage in highly dense areas. Fig. 2 illustrates several scenarios of the UAVs-enabled MEC in the real-time environment. More specifically, Deep Q Network (DQN) based RL mechanism is implemented for handling significant challenges such as energy-efficient trajectory planning in a multi-UAV environment and QoS assurance for GUs. The problem is formulated as a Markov Decision Process (MDP), where the objective is to maximize the total system reward while considering the UAVs' limited energy and GUs QoS constraints. For maximizing the reward, the proposed work uses the greedy-policy approach. With the help of DQN, the UAVs plan their trajectories dynamically by fulfilling the service demands from the GUs. Simulation results reveal that our DQN model outperformed conventional RL algorithms regarding QoS assurance to GUs and convergence rate. It also serves more GUs by optimizing the trajectory.

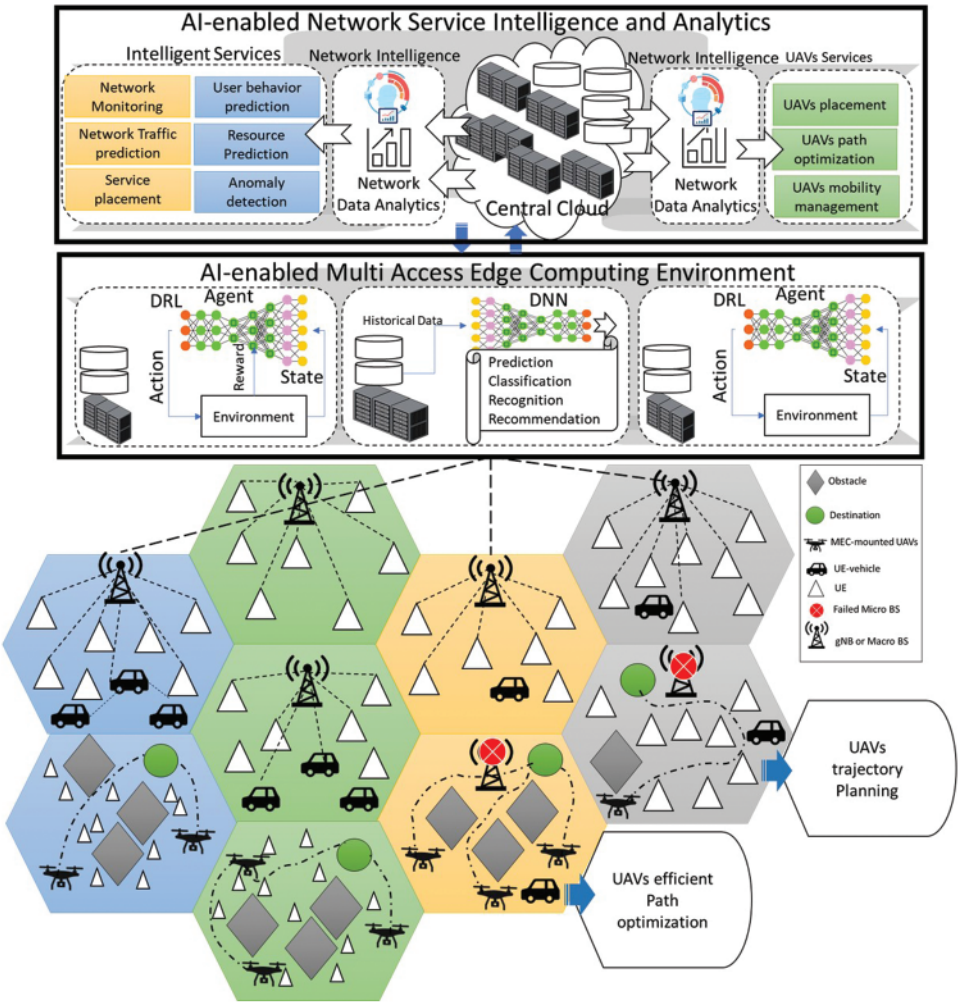

Figure 2: Architecture of proposed DRL-based UAVs-enabled MEC trajectory planning mechanism

The paper is structured as follows. Section 2 explains the related work about AI and DRL approaches for UAVs trajectory planning and service provisioning. The architectural details of the proposed DRL-based system for UAVs trajectory planning are presented in Section 3. Section 4 explains the simulation setup and results achieved through our system. The final section presents the conclusion and future work of the implemented system.

2 Related Work

UAV navigation aims to direct UAVs to the desired destinations on conflict-free efficient routes without human intervention [22]. UAVs are being used by the research community in many applications such as medical [23], agriculture [24,25], food [26], and forestry [27]. It plays an essential role in automated operations under challenging environments. The UAV navigation problem has shown some promise for recent emerging DRL methods [28]. However, most of these approaches do not converge. Guo et al. [29] have developed a DRL framework for UAV automated navigation in a highly dynamic and complex environment. They have specially designed a distributed DRL framework that

decomposes the UAV navigation function into two simple sub-functions. Each sub-task is solved by a DRL network based on long short-term memory (LSTM) created with only a fraction of the interactive data. In addition, they proposed a clipped DRL loss feature that would unite the two sub-solutions to the UAV navigation problem into a single component. The simulation results show that the proposed method is better than the innovative DRL method in terms of convergence and efficiency. The sharing of observation data between multiple UAVs increases the detection range of each UAV. Therefore, this type of research is significant for improving overall mobility further.

Automated management of undefined environments remains a challenge for small UAVs. Recently, several neural network-based technologies have been proposed to solve this problem. However, the trained network is vague, unintelligent, and challenging to understand by humans, limiting its practical application [30]. The study by He et al. [31] discusses the problem of automated flight design of UAVs using DRL. Unlike other studies, the proposed network was trained in the simulation to determine the path of the UAV. They proposed a new way of describing models based on different features to better understand the trained model. Shapley values are used to come up with an explanation method. Textual responses are created using these values, which support the agent's response toward the goal. If the UAV experiences an object in the environment, a textual response is created by Convolutional neural network (CNN) in accordance with that action. This method is then applied to the real-world environment, and a description of the action of UAV was obtained successfully. After an actual flight test is performed, it is concluded that the approach was strong enough to be applied directly to the real environment. However, the model can be improved based on providing more descriptions of the action of the UAV.

By using the powerful MEC servers, computation offloading facilitates the execution of demanding computational workloads. As a result, the quality of computing, such as execution delay enhanced significantly [32]. In work by Khan et al. [33], they proposed an integer linear optimization-based efficient computation offloading technique. For each mobile device, the algorithm offers the options of local execution, offloading execution, and task dropping as the execution modes. The system is based on an enhanced computing method that uses less energy. Lu et al. [34] suggested two secure transmission techniques for multi-UAV-assisted MEC based on single-agent and multi-agent RL. They proposed an approach that starts by optimizing the deployment of UAVs, which covers all users with the fewest possible UAVs, using the spiral placement algorithm. Then, RL is applied to enhance system utility by taking into account various user tasks with varying preferences for processing time and residual energy of computer hardware, which minimized the risk of information eavesdropping by a flying eavesdropper.

In the cases of extensive machine-type communications, UAV-based communication is thought to be a potential remedy for data traffic explosions. The best QoS in UAV-enabled cellular networks is a topic covered in the article by Zhu et al. [35]. An integrated design of access point selection and UAV trajectory planning was proposed to maximize the usability of UAVs. They provided an algorithm that directs the UAV to choose a location with a good channel condition. A game theory-based access point selection algorithm is also implanted, enabling users to select the ideal access point according to a cost function. The algorithm instructs the users to choose an access point automatically because the sub-problem for access point selection is NP-hard. They achieved the best channel quality by using an in-depth strengthening learning DRL approach to enable UAVs to take the best possible action at each location to provide an optimal path for the UAV. According to the simulation results, this method significantly lowers the typical cost to the users. The path design scheme for UAVs can produce a route with a minimum average channel path loss compared to other methods. However, sending more UAVs to larger areas is possible using a multi-agent reinforcement learning framework.

The energy-efficient fair communication trajectory design and frequency band allocation (EEFC-TDBA) algorithm were proposed in the article by Ding et al. [36]. They worked on the path design and frequency band separation of UAVs in a 3D plan to offer fair and energy-efficient communication services. The power consumption model for quad-rotor UAV is presented based on the UAV's 3D motion. The EEFC-TBDA algorithm is implemented based on DRL in which a deep deterministic policy gradient is applied. EEFC-TDBA allows adjusting flight speed and direction to increase energy efficiency and reach the destination before energy depletion. The proposed algorithm also allocates frequency bands and provides a fair communication service. Simulation results were presented to show that EEFC-TDBA performs better in terms of throughput. Within limited energy availability, this approach keeps the balance between the fairness of GUs and throughput. However, the path design and allocation of multiple resources using multiple UAVs must be tested to offer reliable communication to the GU.

Path design for UAVs remains challenging in dynamic environments with potential threats. In the article by Yan et al. [37], they proposed a DRL approach to designing UAV routes based on information in the global context. All the important attributes are selected, such as the location of the UAV, location of goal, and enemy. STAGE Scenario software was also selected to provide a simulation environment for developing quality assessment models that consider the enemy's radar detection and the UAV's viability in a missile attack [37]. A dual depth Q network (D3QN) algorithm is used with a set of quality maps as input to estimate the Q values that fit all candidate behaviors. They combine the greedy and heuristic approach for selecting actions. Furthermore, simulation results are given to show the performance of the proposed approach in static and dynamic work settings, given the information related to the situation. The information related to the situation is not easily available in a real-time scenario. Multiple UAVs can share information related to the global situation for better performance of this approach.

The problem of trajectory planning and resource allocation (TPRA) in multiple drone cells (DCs) is investigated by Shi et al. [38]. The paper considered a large-scale radio access network where multiple base stations are deployed, and the mobility of users in the environment is higher. The authors considered the energy consumption from the UAV to the user and energy consumption from the UAV to the base station. The problem is formulated as MDP, in which the throughput is increased over a huge area. Hierarchical deep reinforcement learning based trajectory and resource allocation (HDRL-TPRA) approach is proposed for DCs in which the high complexity of the environment is solved by dividing the problem into two sub-problems of global trajectory planning (GTP). One subproblem is higher-level GTP, and the other subproblem is low-level GTP. The GTP subproblems aim to do the trajectory planning for multi DCs and increase the number of users covered in a given environment. The GTP is solving the route design over a long period. To solve the sub-problem, they proposed a multi-agent DRL based GTP algorithm to solve the unstable space situation by the environment in which the multi DCs is deployed. Subsequently, every DC independently solves the TPRA sub-issue to control the power distribution, motion, and transmission based on real-time changes in user traffic. Also, the TPRA sub-problem is solved using the heuristic policy gradient algorithm named deep deterministic policy gradient-based lower-level local trajectory planning and resource allocation (DEP-LTPRA). The article concluded with a 40 percent increase in the throughput of the network compared to the existing non-learning-based approach. The high dynamicity of users required a frequent change in the position of drones, leading to high power consumption.

The path planning of UAVs is a compulsory component of rescue operations because UAVs can be used in different types of rescue missions. As UAVs operate in a dynamic, changing environment and high task space, the traditional techniques cannot provide an optimal strategy of control in a

3D environment [39]. Therefore, in the study by Li et al. [40], they proposed a UAV path design system based on DRL that allows UAVs to navigate in a 3D space. The authors assumed a fixed-wing UAV, and a deterministic policy gradient (DDPG) algorithm is proposed in which the UAV takes the decision automatically. In addition, the reward is formulated to navigate the UAV towards the goal while considering its safety. They used a threat function in the reward, enabling the UAV to navigate safely without collision. Simulation results are given, which show that the UAV can navigate safely without collision with obstacles. However, the performance of this approach can be reduced when there is no exact deterministic knowledge about the obstacle and target point.

MEC gathers computer capabilities from the network edge to perform compute-intensive tasks for multiple IoT applications [41]. Meanwhile, UAVs have great flexibility to extend coverage and improve network performance [42]. Therefore, the use of UAVs to provide terminal computing services for large-scale IoT devices is very important. In the article by Peng et al. [43], they study the problem of intelligent path design for UAV-enabled edge networks, considering the mobility of IoT devices. The authors consider the movement of devices in a practical environment using the Gaussian Markov random motion model. Considering the energy consumed during flight and the execution of UAVs, they set up a path design problem to maximize the amount of data that is uploaded to UAVs by the devices. They use the DRL method to build an online path design algorithm based on double deep Q-learning network to handle dynamic changes in complex environments. Extensive simulation results confirm that the proposed algorithm performed better in terms of convergence speed. The data uploaded to the UAV is maximized, and the energy consumption is minimized. However, the performance of the proposed algorithm can be affected when the mobility of the devices is considered.

In the article by Liu et al. [44], they study mobile-edged computer networks equipped with UAVs. In work, UAV performs different computational operations on the data uploaded from the users called terminal users (TUs). All TUs are deployed using a random Gauss-Markov model (GRMM). The Path of the UAV is executed while considering the energy consumption of the UAV. This path design problem is formulated as MDP. In this problem, user association and UAV path are considered for optimization. The authors followed QOS based epsilon greed policy approach to fulfill the QoS and increase the overall reward. The simulation results show better convergence speed as compared to the RL approach. The approach maximizes the throughput and achieves better QoS. However, the algorithm performance can be reduced with the increasing number of Users. Only a single UAV cannot provide the target QoS in a dense environment.

In the cellular coupled UAV network, Li et al. [45] consider the problem of minimizing the time cost and expected time of outage. A UAV navigation approach has been developed using the variable mobility of UAVs to achieve the above optimization objectives. Current offline optimization technology solves the inefficiency in performing UAV path navigation because it realistically considers the distribution of objects, buildings, and placement of antennas directionally. The authors focused on ground air channels and considered objects and buildings with antennas that transmit radiation in 3D. An active solution called quantum-inspired experience reply (QIER), based on DRL is proposed to enable the UAV to find the optimal flight direction. It is concluded that the proposed algorithm needs fewer parameters and is very convenient for implementation. The efficiency and superiority of the proposed solution were shown and validated by numerical results compared to some DRL-related and unlearned solutions.

However, from the cited work, efficient trajectory planning in a multi-UAV environment by preventing UAVs collision and detecting accurate service demand from the GUs is still challenging.

So, the proposed DQN-based model can efficiently plan the trajectory in a multi-UAV environment and assure the different QoS requirements for GUs.

3 Design and Architecture of Proposed DRL-Based UAVs-MEC Trajectory Planning Mechanism

The proposed DRL-based UAVs-enabled MEC framework are illustrated in Fig. 2, which introduces the intelligence at the cloud and edge for efficient service provisioning and management using UAVs networks. In the cloud, the efficient UAVs deployment decision can be made per the requirements such as emergencies, disasters, and to enhance network coverage cases. The network operators (NOs) can proactively manage and control the network resources using AI technologies. These AI algorithms can perform network data analytics such as user behavior prediction, traffic volume prediction, resource consumption, energy efficiency, path planning and optimization for UAVs, mobility management, etc. On the other side, MEC offers cutting-edge capabilities to its customers that are only possible with UAVs. But efficient trajectory planning in the muti-UAVs environment and providing services to the GUs by considering energy efficiency is challenging. So, the proposed model is implemented by using DRL-based mechanism that can efficiently plan the multi-UAVs trajectory by considering the energy and QoS from the GUs. An in-depth discussion is provided on the proposed environmental modeling and optimal trajectory planning in the forthcoming subsections.

3.1 Environment Modeling for DRL-Based Mechanism

The proposed work considers two crucial aspects of MEC-enabled UAVs for planning their trajectories by satisfying the QoS requirements of GUs. It also avoids the collision that occurs between UAVs. Several GUs and UAVs are present in the environment. The UAVs efficiently plan their path to accommodate the GUs demand. Consider each GU has an initial demand for UAVs to process, and the demand can only be serviced when the UAVs are within the GUs' service radius. Due to restricted capabilities, UAVs cannot accurately detect demand signals at a significant distance. As a result, $S_a(P_i, \gamma)$ denotes the service area of a GU, P_i denotes the GU's initial location, and gamma denotes the radius. Fig. 3 shows the environment modeling for the DRL agent, which contains ten GUs with two UAVs.

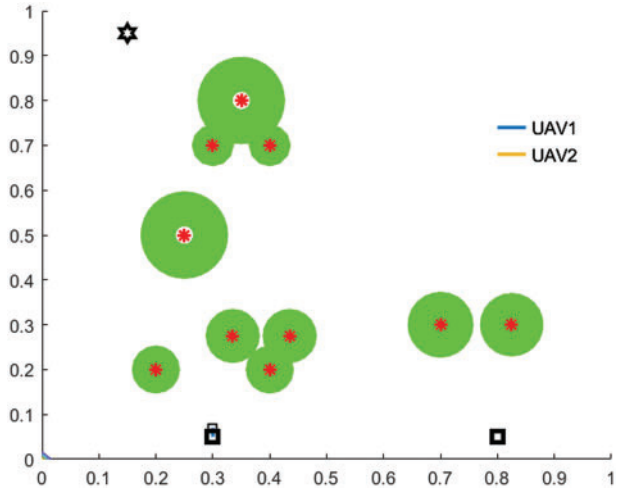

Figure 3: Simulation setup with UAVs and GUs' distribution in a grid environment

Moreover, a GU's service is activated whenever a UAV enters its service area (green circle). The remaining GUs needs will reduce steadily. It can be concluded that a GUs with a higher demand requires more time to be served. So, the GU gets more service if the UAVs stay longer within its service area. The correlation between UAV detected demand and GU demands is not linear. So, the sigmoid demand detection function $U\left(D_{gu}(t)\right)$ is used to accurately detect the GUs demand for assuring better QoS [46]. Eq. (1) illustrates the sigmoid function $U\left(D_{gu}(t)\right)$ for describing the correlation between actual and detected demand, whereas η and β are the constant terms used for controlling the $U\left(D_{gu}(t)\right)$.

$$U\left(D_{gu}(t)\right) = 1 - exp\left[-\frac{D_{gu}(t)^{\eta}}{D_{gu}(t) + \beta}\right] \tag{1}$$

To improve QoS, $U\left(D_{gu}(t)\right)$ can motivate a UAV to concentrate on GUs with higher unserved demand and restrict it from serving any GU for a long period. $U\left(D_{gu}(t)\right)$ increases rapidly as the service demand increases and becomes slow whenever it is sufficient.

A reward function is implemented for UAVs to better learn and adapt in order to choose the optimal Path. This reward function considers the distance and GU service demand. The reward function also calculates the penalty or reward in time period t between two points on the map. The reward function R_{t+1} between two points P_i and P_f is defined in Eq. (2). The coefficient K is used as a service demand that can accommodate the GUs demand as required.

$$R_{t+1} = U\left(D_{gu}(t)\right) + D_{(m,n)} \tag{2}$$

The policy in the RL mechanism refers to the probability of selecting the action A_t as per the current state S_t. So, the main aim is to find the optimal policy π^* to maximize the system's long-term reward as defined in Eq. (3).

$$\pi^* = argmax\frac{\sum_t^{t-1} R_{t+1}}{T} \tag{3}$$

3.2 Proposed DQN for Optimal Trajectory Planning

This study uses the RL method to explore the dynamic and unknown environment. Each UAV in the environment acts as an RL agent where the UAV takes actions to maximize the long-term rewards by trying several actions. UAV agent keeps learning from the feedback and reinforces the actions until the actions yield the optimal result. The agent in DRL constantly trains the DNN based on rewards to optimize its action for a specific state in the environment. Several DRL algorithms proposed in recent years are categorized into on-policy and off-policy [19–21]. Off-policy algorithms, like Q-learning, update the policy by getting a reward for the action through the epsilon greedy technique. The two most notable methods of off-policy models are DQN and Double Deep Q Network (DDQN) [19,21]. Instead of using the Q-table to learn from the rewards and choose the action, DQN employs the two DNN models with the same layers but different parameters. These DNNs models help DQN to learn large and dynamic environments efficiently. Additionally, the proposed work employs the DQN method of the DRL to deal with the large state-action pairs evolved by GUs position and service demand.

In the proposed DQN model, a three-layer DNN model is adopted, which is named as predicted Q_p^n network to estimate state-action pairs Q value at every iteration. Fig. 4 depicts the workflow of the proposed DQN-based mechanism for efficient policy optimization during trajectory planning of UAVs. The input of the Q_p^n model is state information, and the output is the vector of estimated Q

values for all actions of the observed state. The DNN model approximates the Q value using the gradient-decent algorithm in training. The first and second hidden layers contain 128 and 64 neurons, respectively, and Rectified Linear Unit (RELU) is used as an activation function. Moreover, the size of neurons is the same as action space in the output layer, and Softmax is used as an activation function. An additional target network Q_t^n is used to overcome the policy-oscillation issue that occurs due to little change in Q values. The target network Q_t^n has the same settings as predicted network Q_p^n but different parameters. The target network Q_t^n estimates the target Q values. In specific steps, the DQN model shows more stable performance by freezing the parameters of target network Q_t^n.

Algorithm 1: DQN-based policy optimization mechanism for UAVs trajectory planning.

1. Input: $\{R_m, N_e, \epsilon, \theta_1, \theta_2, MAX_s, Min_s, S\}$
2. R_m = replay memory, MAX_s = Maximum R_m size, Min_s = Minimum R_m size, N_e = total number of episodes, ϵ = exploration probability, θ_1 = parameter of Q_p^n, θ_2 = parameter of Q_t^n, S = Step
3. Define Q_p^n with random weights: θ_1
4. Define Q_t^n with θ_1 weights: $\theta_2 = \theta_1$
5. For $E = 1 : N_e$ **do**
6. Select random action A_t with epsilon probability
7. Otherwise $A_t = argmax_A Q_p^n(S_t, A, \theta_1)$
8. Perform A_t obtain agent reward R_t and next state S_{t+1}
9. **if** $R_m == MAX_s$ **then**
10. Delete the oldest values from the memory
11. Store experience $(S_t, A_t, R_t, S_{t+1}, Q_t)$ into the memory
12. **end if**
13. **if** $S > Min_s$ **then**
14. Extract randomly mini batch samples from the R_m memory
15. **end if**
16. **if** $TERMINATE == S_{t+1}$
17. Q_t^n gets R_t
18. **else**
19. $Q_t^n = R_t + \gamma max_{A_{t+1}} Q_t^n(S_{t+1}, A_{t+1})$
20. **end if**
21. Update weights by performing Gradient Descent $\left(Q_t^n - Q_p^n(S_t, A_{t+1}, \theta_1)\right)^2$
22. Update θ_2 to θ_1 at every iteration
23. Update ϵ
24. **end for**
25. **Output:** Optimal policy

The agent observes the state S_t at every time step t and is the input of the predicted network Q_p^n. The agent executes an action A_t by using the epsilon greedy policy as per the output value Q_t, as a result the agent obtains the reward R_t and a new state S_{t+1}. This transition experience $(S_t, A_t, R_t, S_{t+1}, Q_t)$ is stored in replay memory R_m. The maximum and minimum size of replay memory is represented as Max_s and Min_s, respectively. The oldest transition experiences have been deleted whenever R_m becomes full, the latest transitions will be stored. The training process starts only when at least Min_s values are in R_m memory. To effectively train the Q_p^n network, Gradient-descent is adopted to update the parameters θ_1 by using random mini-batch samples from the R_m.

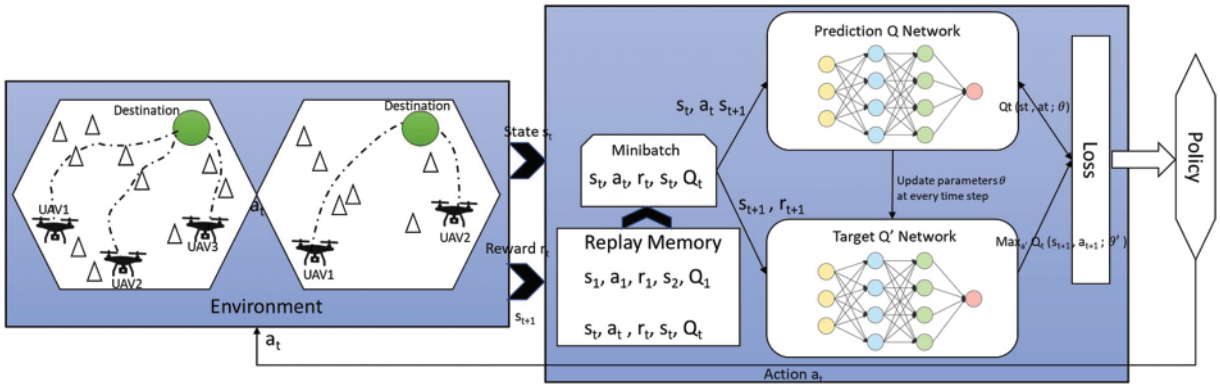

Figure 4: Implemented DQN model structure and procedure for training

In proposed mechanism the decreasing epsilon-greedy policy is adopted for action selection during the training process. The agent acts randomly with epsilon exploration probability and selects the action with maximum Q value having exploitation probability (1-epsilon). The epsilon value decreases with the increased number of training episodes, meaning that the agent gradually shifts from exploration to exploitation. The learning rate γ is set to 0.002. The parameters θ_2 of the target network Q_t^n are frozen in the training phase and are updated from the predicted network Q_p^n by copying θ_1 to θ_2 at every iteration. Finally, the predicted network Q_p^n approximates the action-value function satisfactorily, and optimal policy is achieved by selecting the maximum output value in the current state. Algorithm 1 explains the working procedure of the proposed mechanism.

Algorithm 2: Pseudo code of UAV navigation and service provisioning.

while $\left(sum\left(Sum_{Target}\right) = UAV_{num}\right)$ **do**
  for $i = 1 : UAV_{num}$ **do**
    if $\left(UAV_{pos} - Target\right) \leq step_l \times 10$ **then**
      $Sum_{Target}(i) = 1;$
    else
      loop = loop + 1
    if $\left(UAV_{pos} = Path_i\right)$ **then**
      delete the first line
    end if
    $t_{goal} = PATHi(1, 1{:}2)/N$
    $dis = norm\left(t_{goal} - UAV_{pos}\right)$
    if $dis > step_l$ **then**
      $UAV_{pos} = UAV_{pos} + \left(t_{goal} - UAV_{pos}\right) \times step_l/dis$
    else
      $UAV_{pos} = t_{goal}$
    end if
    for $\left(k = size\left(GU_{info}\right)\right)$
      if $\left(GU_{k1}^{(k,1)} - UAV_{pos}\right) <= Service_R$ & $GU_{k4} = 0)$ **then**

(Continued)

Algorithm 2: Continued

```
19.                 f = max (0.00001, GU_k3 − 1)
20.                 GU_k3 = f
21.                 if GU_k3 < 0.0001 then
22.                     GU_k4 = 1
23.                     count (k) = loop
24.                 end if
25.             end if
26.         end for
27.     end if
28. end for
29. end while
```

Algorithm 2 explains the pseudo code of UAV target servicing. It starts with initializing the variables. The first while loop will continue until all UAVs arrive at the target stopping iterations. If one UAV arrives at the target, no further planning is needed. If the first line has been visited, delete the first line, and turn path from $N \times N$ size to 1×1 size and calculate the distance from the UAV's current position to the next position. Then move one step toward the target goal direction; otherwise, move to the target goal directly. Then update GUs demand matrix; the demand matrix removes all demand from unserved GUs which are located within the service radius. The iteration ends after serving all targets.

4 Experimental Setup and Results

For simulation and obtaining results, we set up MATLAB R2022A version 9.12. The system is tested on Core i7-1160G7 intel 2.20 GHz processor with a windows operating system. Table 1 shows the system requirements to implement test-bed environment for simulations.

Table 1: Test-bed implementation environment

System requirements	Description
Central processing unit (CPU)	Intel Core (TM) i7-1160
RAM	16 GB
Operating system	Windows 10
Simulation tool	MATLAB
Graph toolbox	Version 1.4.0
IDE (Platform)	MATLAB R2022a (V 9.12)

In experiment setup, $N \times N$ grid is used, which contains the GUs and UAVs association. The notations and their values, along with their descriptions, are enlisted in Table 2. The UAVs parameters are adjusted as per each UAV's capabilities. Ten GUs are initialized at randomly distributed locations with two UAVs. The service demand is randomly assigned within the range of [0:10]. Moreover, UAVs hovering, and flying energy are set to 90 and 100 W, respectively. The efficient trajectory planning process using DQN model with only five iterations is depicted in Fig. 5. Where black circle marks represent two UAVs from different locations, red asterisk marks denote GUs, and the green circle

presents the service demand of GUs with service radius. The black asterisk marks show the target point of both the UAVs. Each mission requires both UAVs to fly to the target (asterisk) and provide service to GUs positioned on the map. When GUs are served, the green circles get smaller, indicating a reduction in service demand. The overlap area of service radius demonstrates that the service demands are accumulative. Moreover, the results illustrate that the UAVs can select a low-risk route to service each GU in a highly dynamic environment. It seems that GUs with higher demand is more appealing to UAVs. Both UAVs will reroute to other places with high service demand once the initial service demand has been completed. While planning trajectory by DQN agent, information sharing is beneficial in preventing UAV collisions.

Table 2: Description and values of notations

Sr #	Notation	Description	Value
1	N	map into N × N grid	20
2	N2	map into N2 × N2 grid when calculating the weight matrix	50
3	β	the parameter in the sigmoid service demand function	8
4	GU_{info}	GUs location matrix	-
5	$S_a\ (P_i, \gamma)$	radius within which a GU can be served	0.2
6	$step_l$	UAV one-step length	0.02
8	K	service demand coefficient	1
9	η	the parameter in the sigmoid service demand function	2

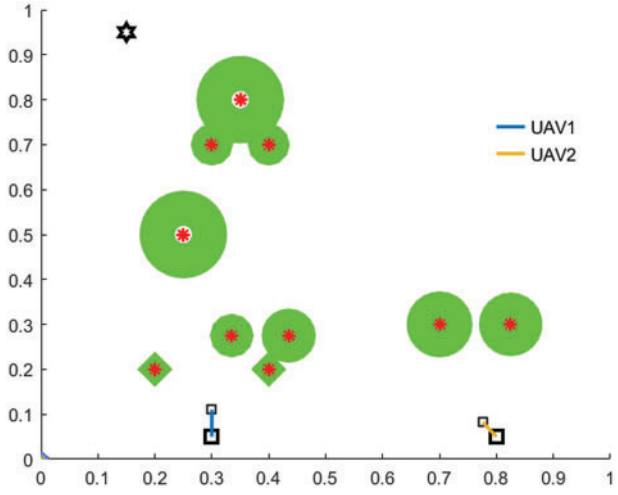

Figure 5: Efficient UAV-enabled MEC trajectory planning by proposed DQN model with 10 iterations

Fig. 6 depicts the DQN trajectory planning process with fifty iterations showing the proposed model's effectiveness for serving GUs with their QoS requirements. Both UAVs planned their trajectory to serve more GUs. Due to UAV collision avoidance capability, both UAVs planned different paths by accommodating more users. As seen, UAV 1 serve four GUs, and UAV 2 serve two GUs by planning their flight. So, the service demand decreases as the UAV provides service to a specific GU. Whenever the UAVs enter the GU service radius, the service begins. As shown in Fig. 6, the green circle representing service demand of GU adopts the shape of small green diamond after being served.

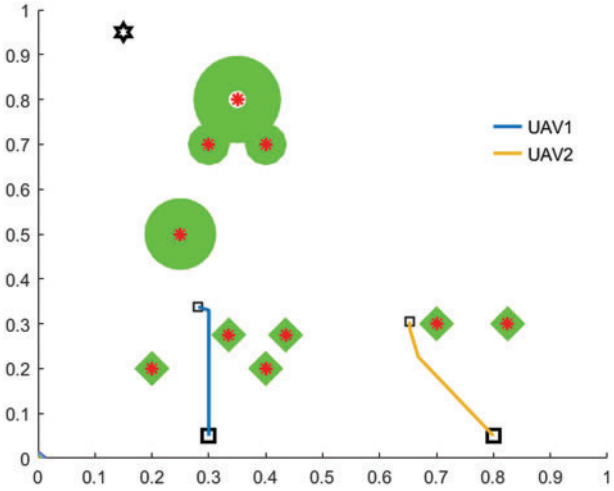

Figure 6: Efficient UAV-enabled MEC trajectory planning by proposed DQN model with iterations = 50

Fig. 7 illustrates trajectory planning procedure of the proposed RL model with 200 iterations where both the UAVs reached at target point after serving all the GUs. All GU's service demands have served, and green circles become small diamonds. The mission accomplishment shows the effectiveness of the proposed mechanism. An efficient trajectory planning is performed and the maximum number of GUs are served with their QoS requirements. Both the UAVs follow the service demand detection from GUs for the planning path.

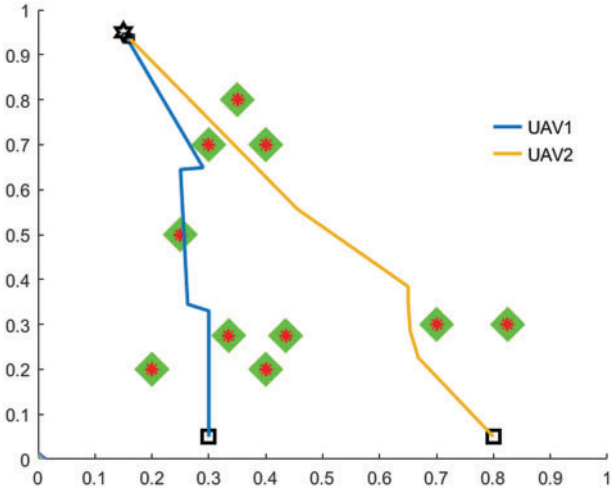

Figure 7: Efficient UAV-enabled MEC trajectory planning by proposed DQN model with iterations = 200

Fig. 8 illustrates the comparison between the proposed QoS aware greedy policy with the conventional greedy policy approach for QoS satisfaction. It can be seen in the graph that the proposed approach achieved almost 99% QoS fulfillments for ten GUs, and in the case of the conventional greedy approach maximum of 50%, QoS is achieved. This evaluation shows the superiority of the proposed

QoS-aware DQN model with simple conventional policy. It also shows the optimal reward shaping and learning of the proposed DQN model.

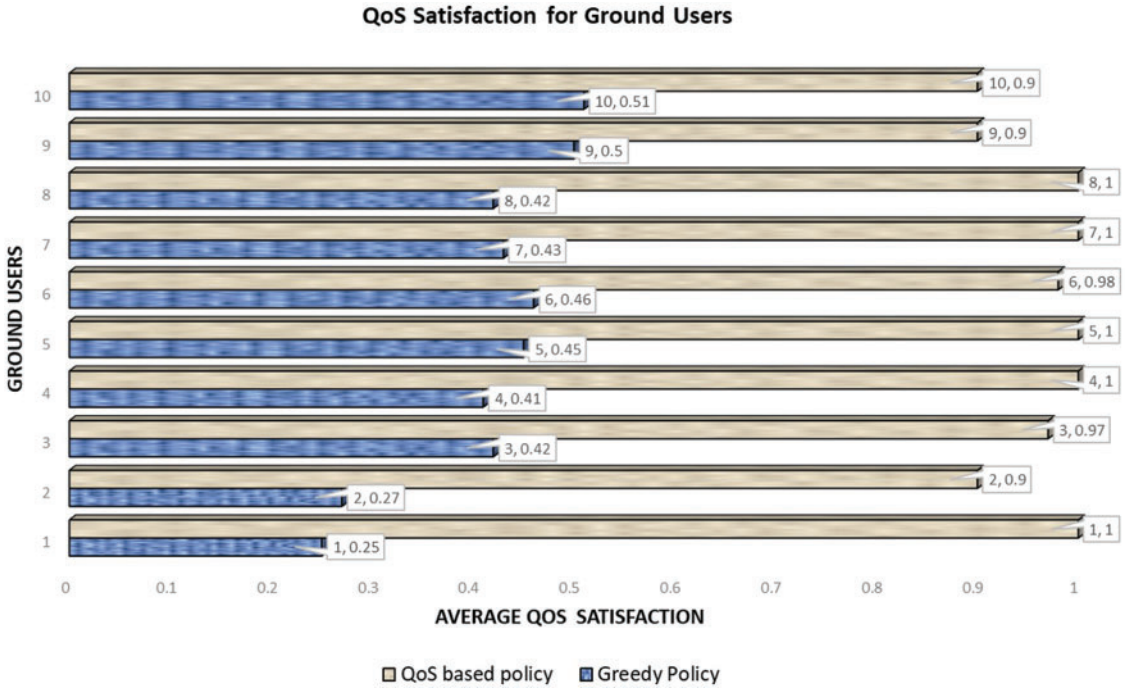

Figure 8: QoS comparison of proposed DQN based QoS-aware greedy policy with conventional greedy policy approach

The results show the superiority of our proposed DQN model for QoS assurance and adaptation of a dynamic environment. Additionally, it shows effective performance in terms of stability and mission completion. The viability and efficacy of the proposed method is obvious. Moreover, UAVs also planned their flight by DQN agent within the energy constraints. So, the proposed mechanism is an effective and optimized solution for energy-efficient UAVs trajectory planning and QoS fulfillment.

5 Conclusion

In this work, proactive trajectory planning and management issues are investigated for UAVs-enabled MEC network while considering the QoS requirements of the GUs. A DRL-based DQN model has been implemented for optimal trajectory planning that aims to fulfill the QoS demands from GUs in an energy-efficient way. The GUs are randomly distributed, and UAVs are tasked to serve them as per their QoS. DQN agents plan the trajectory according to the service demand detected from the GUs. Finally, UAVs plan their trajectories by preventing the same path, collision and providing service to the GUs. Simulation results validate the efficacy of the proposed mechanism in terms of mission completion and stability in trajectory planning for multi-UAVs. This mechanism not only assures the QoS for users but also completes their mission according to energy efficiency of each UAV. In the future, we intend to enhance the proposed work by using advanced algorithms such as DDQN and asynchronous-advantage actor-critic (A3C) models for optimal resource allocation in a UAV-enabled MEC environment. Moreover, we will consider a more complex environment by considering a real map and different weather conditions with real-time QoS demands from the users.

Funding Statement: This work was supported by the Fundamental Research Funds for the Central Universities (No. 2019XD-A07), the Director Fund of Beijing Key Laboratory of Space-ground Interconnection and Convergence, and the National Key Laboratory of Science and Technology on Vacuum Electronics.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] G. Carvalho, B. Cabral, V. Pereira and J. Bernardino, "Edge computing: Current trends, research challenges and future directions," *Computing*, vol. 103, no. 5, pp. 993–1023, 2021.

[3] P. Ranaweera, A. D. Jurcut and M. Liyanage, "Survey on multi-access edge computing security and privacy," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 1078–1124, 2021.

[4] K. Abbas, T. A. Khan, M. Afaq, A. Rafiq, J. Iqbal *et al.,* "An efficient SDN-based LTE-WiFi spectrum aggregation system for heterogeneous 5G networks," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, pp. 3943, 2022.

[5] K. Abbas, T. A. Khan, M. Afaq and W. C. Song, "Network slice lifecycle management for 5 g mobile networks: An intent-based networking approach," *IEEE Access*, vol. 9, pp. 80128–80146, 2021.

[6] K. Abbas, T. A. Khan, M. Afaq, A. Rafiq and W. C. Song, "Slicing the core network and radio access network domains through intent-based networking for 5G networks," *Electronics*, vol. 9, no. 10, pp. 1710, 2020.

[7] J. Ren, G. Yu, Y. He and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[8] S. D. Shah, M. A. Gregory and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10903–10924, 2021.

[9] A. Rafiq, M. S. A. Muthanna, A. Muthanna, R. Alkanhel, W. A. M. Abdullah *et al.,* "Intelligent edge computing enabled reliable emergency data transmission and energy efficient offloading in 6TiSCH-based IIoT networks," *Sustainable Energy Technologies and Assessments*, vol. 53, pp. 102492, 2022.

[10] Q. Luo, S. Hu, C. Li, G. Li and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2131–2165, 2021.

[11] Y. Qian, "Unmanned aerial vehicles and multi-access edge computing," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 2–3, 2021.

[12] Y. Xu, T. Zhang, Y. Liu, D. Yang, L. Xiao *et al.,* "UAV-assisted MEC networks with aerial and ground cooperation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7712–7727, 2021.

[13] N. Fatima, P. Saxena and M. Gupta, "Integration of multi access edge computing with unmanned aerial vehicles: Current techniques, open issues and research directions," *Physical Communication*, vol. 52, pp. 101641, 2022.

[14] Z. Liu, Y. Cao, P. Gao, X. Hua, D. Zhang *et al.,* "Multi-UAV network assisted intelligent edge computing: Challenges and opportunities," *China Communications*, vol. 19, no. 3, pp. 258–278, 2022.

[15] A. Khan, J. Zhang, S. Ahmad, S. Memon, H. A. Qureshi *et al.,* "Dynamic positioning and energy-efficient path planning for disaster scenarios in 5G-assisted multi-UAV environments," *Electronics*, vol. 11, no. 14, pp. 2197, 2022.

[16] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao *et al.,* "Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–343, 2020.

[17] Z. Song, X. Qin, Y. Hao, T. Hou, J. Wang *et al.,* "A comprehensive survey on aerial mobile edge computing: Challenges, state-of-the-art, and future directions," *Computer Communications*, vol. 191, pp. 233–256, 2022.
[18] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. C. Chen *et al.,* "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020.
[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. London, England: MIT press, 2018.
[20] Y. Lecun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[21] A. T. Azar, A. Koubaa, N. A. Mohamed, H. A. Ibrahim, Z. F. Ibrahim *et al.,* "Drone deep reinforcement learning: A review," *Electronics*, vol. 10, no. 9, pp. 999, 2021.
[22] A. Jaimes, S. Kota and J. Gomez, "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles UAVs," in *Proc. IEEE SoSE*, Monterey, California, USA, pp. 1–6, 2008.
[23] S. I. Khan, Z. Qadir, H. S. Munawar, S. R. Nayak, A. K. Budati *et al.,* "UAVs path planning architecture for effective medical emergency response in future networks," *Physical Communication*, vol. 47, pp. 101337, 2021.
[24] P. W. Khan, Y. C. Byun and M. A. Latif, "Clifford geometric algebra-based approach for 3D modeling of agricultural images acquired by UAVs," *IEEE Access*, vol. 8, pp. 226297–226308, 2020.
[25] P. W. Khan, G. Xu, M. A. Latif, K. Abbas and A. Yasin, "UAV's agricultural image segmentation predicated by clifford geometric algebra," *IEEE Access*, vol. 7, pp. 38442–38450, 2019.
[26] S. Y. Lee, S. R. Han and B. D. Song, "Simultaneous cooperation of refrigerated ground vehicle (RGV) and unmanned aerial vehicle (UAV) for rapid delivery with perishable food," *Applied Mathematical Modelling*, vol. 106, pp. 844–866, 2022.
[27] R. Neuville, J. S. Bates and F. Jonard, "Estimating forest structure from UAV-mounted LiDAR point cloud using machine learning," *Remote Sensing*, vol. 13, no. 3, pp. 352, 2021.
[28] C. H. Liu, X. Ma, X. Gao and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1274–1285, 2020.
[29] T. Guo, N. Jiang, B. Li, X. Zhu, Y. Wang *et al.,* "UAV navigation in high dynamic environments: A deep reinforcement learning approach," *Chinese Journal of Aeronautics*, vol. 34, no. 2, pp. 479–489, 2021.
[30] S. Y. Choi and D. Cha, "Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art," *Advanced Robotics*, vol. 33, no. 6, pp. 265–277, 2019.
[31] L. He, A. Nabil and B. Song, "Explainable deep reinforcement learning for UAV autonomous navigation," *Aerospace Science and Technology*, vol. 118, pp. 107052, 2021.
[32] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no no. 3, pp. 1628–1656, 2017.
[33] P. W. Khan, K. Abbas, H. Shaiba, A. Muthanna, A. Abuarqoub *et al.,* "Energy efficient computation offloading mechanism in multi-server mobile edge computing—an integer linear optimization approach," *Electronics*, vol. 9, no. 6, pp. 1010, 2020.
[34] W. Lu, Y. Mo, Y. Feng, Y. Gao, N. Zhao *et al.,* "Secure transmission for multi-UAV-assisted mobile edge computing based on reinforcement learning," *IEEE Transactions on Network Science and Engineering*, early access, Jun. 22, 2022. https://doi.org/10.1109/TNSE.2022.3185130.
[35] S. Zhu, L. Gui, N. Cheng, F. Sun and Q. Zhang, "Joint design of access point selection and path planning for UAV-assisted cellular networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 220–233, 2020.
[36] R. Ding, F. Gao and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7796–7809, 2020.
[37] C. Yan, X. Xiang and C. Wang, "Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 98, no. 2, pp. 297–309, 2020.

[38] W. Shi, J. Li, H. Wu, C. Zhou, N. Cheng *et al.,* "Drone-cell trajectory planning and resource allocation for highly mobile networks: A hierarchical DRL approach," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9800–9813, 2021.

[39] C. Qu, W. Gai, M. Zhong and J. Zhang, "A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning," *Applied Soft Computing Journal*, vol. 89, pp. 106099, 2020.

[40] Y. Li, S. Zhang, F. Ye, T. Jiang and Y. Li, "A UAV path planning method based on deep reinforcement learning," in *Proc. IEEE USNC-CNC-URSI*, Montreal, QC, Canada, pp. 93–94, 2020.

[41] M. Aazam, S. Zeadally and E. F. Flushing, "Task offloading in edge computing for machine learning-based smart healthcare," *Computer Networks*, vol. 191, pp. 108019, 2021.

[42] A. I. Alshbatat and L. Dong, "Cross layer design for mobile adhoc unmanned aerial vehicle communication networks," in *Proc. ICNSC*, Chicago, IL, USA, pp. 331–336, 2010.

[43] Y. Peng, Y. Liu and H. Zhang, "Deep reinforcement learning based path planning for UAV-assisted edge computing networks," in *Proc. IEEE WCNC*, Nanjing, China, pp. 1–6, 2021.

[44] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding *et al.,* "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5723–5728, 2020.

[45] Y. Li and A. H. Aghvami, "Intelligent UAV navigation: A DRL-QiER solution," in *Proc. ICC 2022-IEEE Int. Conf. on Communications*, Seoul, South Korea, pp. 419–424, 2022.

[46] J. W. Lee, R. R. Mazumdar and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the internet," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 827–840, 2005.