

Article

Deep Reinforcement Learning Based Resource Management in UAV-Assisted IoT Networks

Yirga Yayeh Munaye ^{1,*}, Rong-Terng Juang ², Hsin-Piao Lin ³, Getaneh Berie Tarekegn ¹ and Ding-Bing Lin ⁴¹ Department of Electrical Engineering and Computer Science, National Taipei University of Technology, Taipei 10608, Taiwan; gechb21@gmail.com² Department of Electronic Engineering, Feng Chia University, Taichung 40724, Taiwan; rtjuang@mail.fcu.edu.tw³ Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; hplin@mail.ntut.edu.tw⁴ Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; dblin@mail.ntust.edu.tw

* Correspondence: byyirga@gmail.com; Tel.: +886-901-292-467

Abstract: The resource management in wireless networks with massive Internet of Things (IoT) users is one of the most crucial issues for the advancement of fifth-generation networks. The main objective of this study is to optimize the usage of resources for IoT networks. Firstly, the unmanned aerial vehicle is considered to be a base station for air-to-ground communications. Secondly, according to the distribution and fluctuation of signals; the IoT devices are categorized into urban and suburban clusters. This clustering helps to manage the environment easily. Thirdly, real data collection and preprocessing tasks are carried out. Fourthly, the deep reinforcement learning approach is proposed as a main system development scheme for resource management. Fifthly, K-means and round-robin scheduling algorithms are applied for clustering and managing the users' resource requests, respectively. Then, the TensorFlow (python) programming tool is used to test the overall capability of the proposed method. Finally, this paper evaluates the proposed approach with related works based on different scenarios. According to the experimental findings, our proposed scheme shows promising outcomes. Moreover, on the evaluation tasks, the outcomes show rapid convergence, suitable for heterogeneous IoT networks, and low complexity.

Keywords: wireless resource management; deep reinforcement learning; unmanned aerial vehicles; wireless networks



Citation: Munaye, Y.Y.; Juang, R.-T.; Lin, H.-P.; Tarekegn, G.B.; Lin, D.-B. Deep Reinforcement Learning Based Resource Management in UAV-Assisted IoT Networks. *Appl. Sci.* **2021**, *11*, 2163. <https://doi.org/10.3390/app11052163>

Academic Editor: Sunghun Jung

Received: 30 December 2020

Accepted: 25 February 2021

Published: 1 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

1. Introduction

Recently, the development of IoT applications, the consideration of efficient wireless service for IoT devices, and the issue of resource management (RM) become vital. The overall performance of RM involves the efficient and dynamic use of resources such as times, bandwidth, and frequency [1]. Therefore, higher throughput, higher data rate, lower interference, and better coverage are appropriate considerations for RM in the IoT-based wireless networks. The unmanned aerial vehicles (UAVs) are used in large-scale applications such as security inspection, aerial patrol, and traffic assessment [2]. Hence, UAV-assisted resource management becomes vital for the advancement of the fifth-generation networks. The reasons for the use of UAVs to assist the resource management include: (i) it can be used for rapid management of resource requests from the overloaded IoT users; (ii) it is easy to accommodate the further growth of system coverage and capacity; (iii) it can be operated at different altitudes; and (iv) it can provide rapid and on-demand service for IoT users [3,4]. Due to the higher altitude and greater coverage, aerial base stations have a higher chance of having line-of-sight (LOS) connections with ground users. Therefore, to provide reliable services to IoT users, UAV based resource management is an essential and hot issue for enhancing the resource utilization of IoT networks [5,6].

In order to improve the accuracy of regression and classification tasks in the RM scheme, deep learning methods are more preferable over conventional machine learning methods [7]. Additionally, deep learning methods allow extracting features automatically from large datasets [8,9]. Specifically, deep reinforcement learning (DRL) has made a substantial improvement in RM that is difficult to model with conventional approaches [5]. Mainly, conventional methods have faced a significant challenge due to the wide range and complexity of wireless networks [10]. To this end, the gaps of traditional machine learning-based RM schemes, such as the model complexity, costly use training, and the generalization from test workloads to the actual application of user workloads are identified. Therefore, to solve the most complicated RM problems in IoT networks, this paper proposes a DRL method, in which the state, action, and reward are important parameters that should be designed to generate an optimal policy [11]. The DRL-based approach has been applied to various fields, such as resource management and allocation [12,13], dynamic channel access [12], mobile offloading [13], mobile and unified edge computing caching, and communication (3C) [14–16], fog radio access networks [17,18]. The implementation of DRL is more successful than the single agent Q-learning approach [19]. However, most of them were limited to designing and analyzing the DRL-based method in fixed base stations for solving the joint resource allocation problems. As a result, our proposed method can solve the joint RM optimization by using the DRL approach. According to Reference [20], the DRL agents consider the maximum long-term rewards rather than simply obtaining the current optimal rewards. This is critical for time-changing and dynamic systems.

This paper focuses on UAV-based RM with the application of the DRL approach. Our main initiatives are applying a multi-agent-based DRL model, round robin with K-means for user request queue, and clustering tasks. To the best of our knowledge, these issues are open ones that have not yet been fully investigated. To jointly manage user data, as well as UAV-based orthogonal frequency-division multiplexing (OFDM) signal values in a real environment, the DRL approach is applied. We considered one UAV connected through wireless backhauls with the core network. Under this layout, the objective is to optimize the management of resources for IoT users based on the A2G access links. Different from previous studies, our study focuses on the actual data collection environment. Additionally, this paper uses a large dataset with the integration of DRL and K-means. The real data collection includes both the signal variations and their LOS/ non-line of sight (NLOS) views. Thus, the DRL can reduce the computational times and is a better way to test a large number of datasets, and produce an optimal strategy with a range of real environments relative to conventional RM approaches [21]. Finally, it inspired us to investigate the issues based on the application of the DRL approach. Figure 1 illustrates the general architecture of the DRL algorithm for our system design.

The architecture displays both the primary and target networks, which contains all the obtained dataset values and the final results for allocating resources, respectively. Then, the weights of DQN are outlined to update the loss values based on the primary and target networks. Additionally, the action, state, and reward section values are considered in the environment. Then, all these are updated in the next action, next state and next reward, respectively. In the end, all values are stored into replay memory and used for making policies and decisions. Therefore, the proposed DRL learning architecture can learn user data from the grid points, clusters, and UAV altitudes. As the proposed learning architecture can automatically learn the characteristics of the environment based on the learning input sequences with different time scales.

The major contributions of our work can be summarized as follows: We consider a UAV-assisted wireless IoT network and assess the RM schemes. Then, we propose a multi-agent DRL with a round robin resource-scheduling algorithm for the optimization of joint RM. A joint RM optimization method is proposed to minimize the power consumption and maximize the user throughput and the signal-to-interference-plus-noise ratio (SINR)

- (1) We design a system model that is connected to IoT users, UAV-BS, and A2G channel access links. This framework contains a DRL based RM problem with multiple constraints, such as the number of users, channel gains, signal noise ratio (SNR) issues, and power consumption levels. These variant parameters are used to characterize the heterogeneous and dynamic nature of the environment at each time slot.
- (2) The DRL is applied for the development of the main system model with K-means as a clustering approach. Then, the round-robin algorithm is used to handle the service request queue for the IoT users. The IoT devices are clustered into cluster 1 (urban) and cluster 2 (sub-urban) based on their location and signal distribution. This makes our system more computationally efficient and stable. The proposed DRL framework is therefore used to perform the optimal RM for the UAV assisted IoT devices. Additionally, the DRL techniques with neuron activation mechanisms are used to compare and evaluate the impact of neuron activation on the convergence of the proposed system.
- (3) Ultimately, based on the proposed system, two scenarios (cluster 1 and cluster 2) are used for system evaluation. It is important to handle mobile users within the transmission range of the A2G based UAV access link.

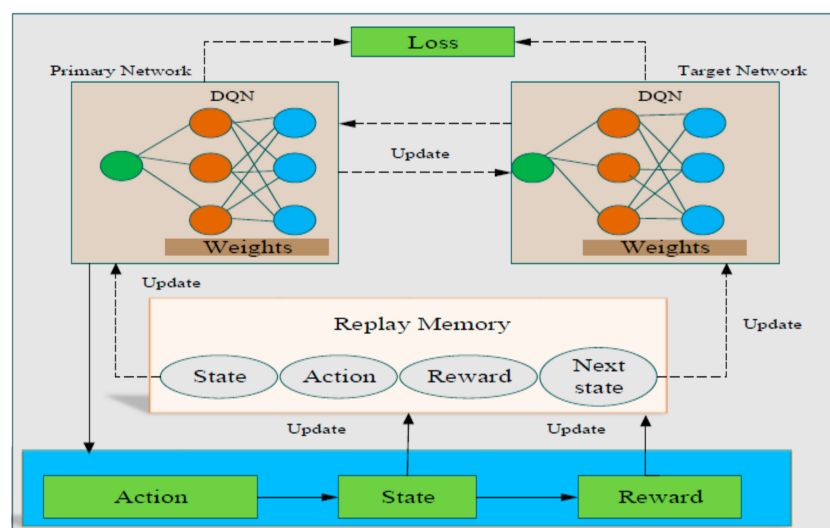


Figure 1. Overall architecture of deep reinforcement learning.

In the rest of the paper, Section 2 provides the related works, and Section 3 explains the system model architecture. In the Methods and Materials Section, such as experimental setup and data collection procedures, are discussed in Section 4. Section 5 provides the proposed system design, such as the round robin scheduling-algorithm for resource management, multi-agent DRL for joint resource management, proposed system architecture, and evaluation metrics. Section 6 reports the experimental results and discussions. Finally, the conclusion and future works are outlined in Section 7. Lastly, references are listed.

2. Related Works

In the wake of IoT heterogeneous networks, there is a need for equal management of network resource strategies. The DRL-based approach has been used to resolve various issues related to resource management and allocation [22]. However, the objectives are highly limited to the analysis of the DRL-based approach. In Reference [23], the reinforcement learning approach has been suggested for user group vehicle networks. The objective was to investigate the optimal power control solution and network capability in heterogeneous networks (HetNets) based on the resource management strategy. In previous studies, Q-learning has been extensively applied as the reinforcement learning approach [24]. Additionally, based on the Q-learning approach, it is difficult to handle

the large environment [25]. The action is also limited for small section optimal approach conditions. However, none of the works centered on the HetNets A2G UAV-based access links. Hence, DRL [23] is assumed to be an effective method for resolving the complex and joint RM system. More specifically, it is applicable for controlling the fluctuation effects of the single-agent framework based on the resource sharing system [19,26]. The authors investigated the reinforcement learning-based approach to improve the strategy for power control and network rate adaptation. In Reference [26], the initial users and secondary users based on a constant control were considered. The objective was to adjust the power based sharing scheme after the detailed learning. Similarly, Reference [27] suggested a new approach of centralized DRL-based power allocation. The authors used the deep Q-learning method to perform near-optimal power allocation and to achieve a higher user throughput. Also, Reference [28] suggested a DRL-based throughput maximization scheme in small cell networks. In this paper, a DRL algorithm with a deep recurrent neural network was used. The channel selection and fractional spectrum access are considered as resources to be managed. However, the existing works do not consider the use of the actual dataset and focus only on the request queue of the user service. In short, the existing Q-learning method works for a limited environment, action, and decision or reward. It is difficult for a large environment to manage the optimal solution with the Q-learning method [27–29].

According to Reference [30], Q-learning and DQN methods were used for the power allocation scheme. The key objective was to reduce interference and cooperation of power allocation to boost the quality of service with LTE-femtocells. Using the same proposed method, the analysis in Reference [20] considered a distributed technique under centralized training conditions. However, the research sought to test outcomes with a fixed allocation of resources. Therefore, at this time, UAVs are used to assist the terrestrial base stations [31] for many purposes, such as power allocation, connectivity enhancement, and throughput maximization [2,20,29]. UAVs can be applicable for RM on-demand standard service [21,32] of power and user throughput as a supplier of information from A2G to IoT devices [22,25]. We, therefore, use UAVs as the key base station to collect data, optimize throughput, and manage SINR. The paper in Reference [33] proposed a multi-agent DRL as resource allocation for vehicle-to-vehicle communications. The system mapped the local observations of channel state information and the interference management level. The study considered vehicles as agents to interact with the environment for efficient power transmission decisions. The objective was to minimize transmission overhead with available resources. Additionally, the authors in References [34,35] intended to use a deep learning-based method for the optimization of wireless resources. The main objective was to forecast a resource allocation based scheme with 24-hour wireless resource management of multiple timescale features.

Unlike conventional resource management methods, such as heuristics-based, game-theoretical, and cooperative approaches, the DRL can derive actions from the run-time context information [3,31]. Therefore, the DRL makes the decision based on retuning and retraining via the distributed and dynamically changing of the IoT environment for automatic resource management [35,36]. While the above works applied DRL for RM, most of them used too small and simulated dataset representations. After critically identifying the gaps, we are initiated to assess the optimization ways of resource management based on various performance indicators. Then, to rapidly analyze and optimize the cluster-based resource management performance through a large number of real datasets, this study focuses on cluster-based data collection with IoT users and UAV based communication. Therefore, our work's unique feature is using a real data collection environment rather than simulations. Then, the K-means method is applied to cluster the data collection environment. Then, we apply a round robin algorithm to manage the resource request queue for the users to generate a service. Then, we apply DRL approaches for the main system model construction. Therefore, our proposed system has the benefit of extracting the most representative features from datasets better than conventional approaches.

3. System Model

Figure 2 displays the UAV-assisted IoT system model. It consists of an A2G signal access link transmission, LOS/NLOS views, and the altitude between the UAV and the user (D_{um}). Assume that UAV is used by ground users to provide access-sharing information. The UAV as a base station is used and deployed at altitudes of 30 and 60 m and reach all the users randomly distributed on the ground inside the actual area. The OFDM signal, as the received signal strength (RSS) and SNR values, which could influence any user equipment, has been collected. The transmitter (Tx), i.e., the UAV, is represented as M and the receiver (Rx), i.e., the mobile user, is represented as U. The system architecture configuration covers the network location with 35 users, denoted by $U1, U2, U3, \dots, U35$, serviced by UAV.

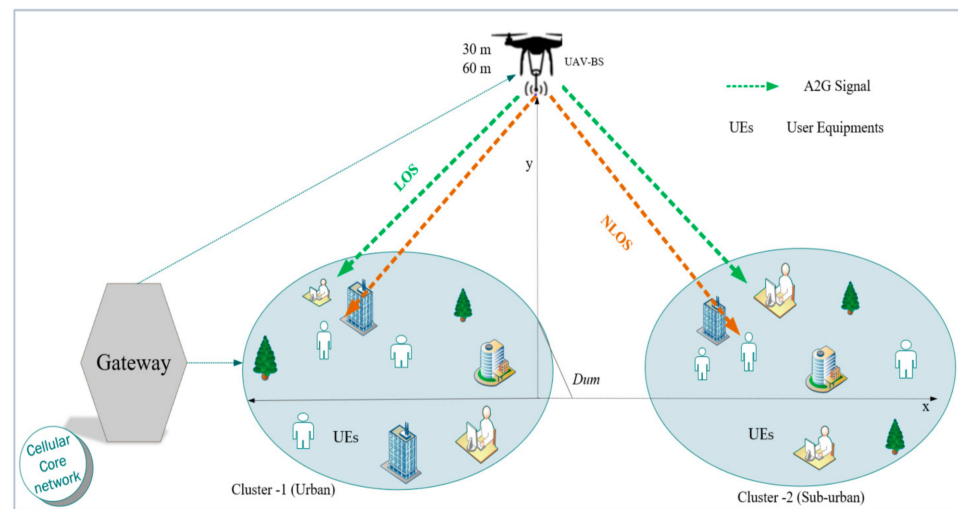


Figure 2. System model: UAV-Bs: unmanned aerial vehicle as base stations; A2G: air-to-ground.

Figure 2 shows the relation between the M-U A2G link; clusters are used as target areas. All mobile users operate on the same frequency band. Therefore, all users are expected to receive the same OFDM signals. The control station is used to monitor and access the network. The defined clusters are presumed to be urban and sub-urban for cluster 1 and cluster 2, respectively. Then, the deployed UAV reaches both clusters based on their altitude and user interest for resource usage. The path loss (P_L) model in the states as S=LOS or S=NLOS is shown in Equation (1) according to References [37,38],

$$P_Ls(D_{um}) = 29.5 \log(D_{um}) + 3.5 + 20 \log(f_c) + \xi_s, \quad (1)$$

where D_{um} is the altitude between UAV and the user (in m), α is the path loss exponent, ξ models the shadowing effect, which is a Gaussian random variable, $N(0, \sigma_s^2)$, while f_c is the carrier frequency. Then, the channel gain is composed of the $P_Ls(D_{um})$ and $S(D_{um})$, which are the path loss and small-scale fading coefficient between UAV to a user, respectively. Then, the channel gain can be expressed, as in Equation (2),

$$C(D_{um}) = P_Ls(D_{um})S(D_{um}), \quad (2)$$

To estimate the interference, it is assumed that all N channel resources are occupied by M to users [26,27]. The UAV at the altitude of 30 m and 60 m has an estimation of SINR on the channel n, which is given by:

$$SINR_C^n = \frac{P_t^1 C(D_{um}) 1 P_Ls(D_{um})}{\sigma_s^2 + \sum_n D_{um} P_t^1 C(D_{um}) 1 + I_{n, N}}, \quad (3)$$

where Pt^1 and $C(D_{um})$ are the power transmission and the channel gain ratio from M to U1 to the control station, respectively; $I_{n, N}$ is the noise power, D_{um} is the altitude from M to Un. Therefore, the overall SINR is expressed as follows:

$$SINR_{Mu}^n = \frac{Pt^1 C(D_{um})}{\sigma_s^2 + \sum_n D_{um} Pt^1 C(D_{um})n + Pt^{D_{um}} G^{D_{um}} + I_{n, N}}, \quad (4)$$

where $C(D_{um})$ and $C(D_{um})n$ are the channel gains between M to U and M to Un based on the altitude of the Tx and Rx, respectively; $Pt^{D_{um}}$ is the power transmission between M to U.

Assuming that there is a physical capacity of Phy_u and user throughput T_U . Then, there is a user throughput capacity, where T_U has an effective throughput of T_{eu} , M_U is the number of users who interact with UAV. Then, T_s is the time slot of a user communicates with (D_{um}) to use the A2G link. The resource request (R_U) ratio of the user, then the maximum T_U , are analyzed in Equation (5). Therefore, the total throughput is measured at a grid level with clusters. To calculate the entire throughput of the separate user and total throughput (T) as in Equation (6):

$$T_U < Phy_u, T_{eu} = t_u T_U, \forall u \in (M_U), T_U = \min(R_U, T_{eu}), \quad (5)$$

$$T = \sum u \in Nu T_u, T = \sum Rss \in Rss_n T_u \frac{Rss_1 + Rss_2 + Rss_3 \dots Rss_n}{Rss_{nc}}, \quad (6)$$

$$T = \sum Rss \in Rss_n T_u \frac{AvgRss_c}{\frac{T_s}{s}},$$

where $Rss_1 + Rss_2 + Rss_3 \dots Rss_n$ is the sum of all Rss_s values in each cluster, $AvgRss_c$ is the average number of Rss_s values per clusters, and T_s/s is the time slot to produce signal values. Therefore, this manages the optimal allocation of throughput values.

Equation (7) calculated the analysis of power transmission (Pt) for the A2G access link with the altitude between M to U (D_{um}). Let $Phy_{D_{um}}$ represent the data rate access ratio from M to U access. As a reference, we used the Shannon formula, as in Equation (8), where G_{Tx} and G_{Rx} are the Tx and Rx antenna gains, respectively, i.e., the difference between the average transmitted and received power in a random transmit as well as received direction. Then, P_t is computed as:

$$Pt_{Dum}(dB) = P_t G_{Tx} G_{Rx} \left(\frac{\lambda}{4\pi D_{um}} \right)^\alpha, Phy_{Dum} = B \log_2 \left(1 + \frac{P_t G_{Tx} G_{Rx}}{I_{n, N}} \right) \left(\frac{\lambda}{4\pi D_{um}} \right)^\alpha, \quad (7)$$

$$Pt_{Dum}(dB) = P_{Tx} + G_{Tx} + G_{Rx} + 10 \log_{10} \left(\frac{\lambda}{4\pi D_{um}^2} \right), \quad (8)$$

where Pt_{Dum} is the full transmitting power of the UAV and the total received power of users by dBm in all directions. The value of lambda (λ) is the wavelength of the radio rate. In short, user throughput is generated through the available values of bandwidth, data rate ratio, and the efficient throughput capacity to users.

4. Methods and Materials

This section describes the data collection methods, such as experimental setup, data collection procedures, and parameter descriptions.

4.1. Experimental Setup

The data collection area (i.e., National Technology University of Technology (NTUT)) is 500 and 700 m as a target of two clusters. The bandwidth and carrier frequency used is 3 MHz and 900 MHz, respectively. First, in a random form of the RM method, the agent randomly selects a sub-band for transmission at each time slot. The users are grouped by their similarities, and sub-bands are allocated iteratively to the user in each grid point. The data were obtained from a horizontal distance of 2×2 m with 35 users per grid of

10 second intervals. As shown in Figure 3a, a UAV-Bs were placed on top of buildings numbered 15 (fifteen academic buildings) and 3 (complex building). The buildings, device users, trees, and other shadowing effects are present in the working environment. The UAV-Bs reached the specified clusters and assumed the central placement for the UAV. An entire raw record of 46,000 values was collected. After preprocessing is carried out, we had a total of 45,500 structured data points. Then, 75% (34,125) and 25% (11,375) of the dataset were selected for the training and testing phases, respectively.

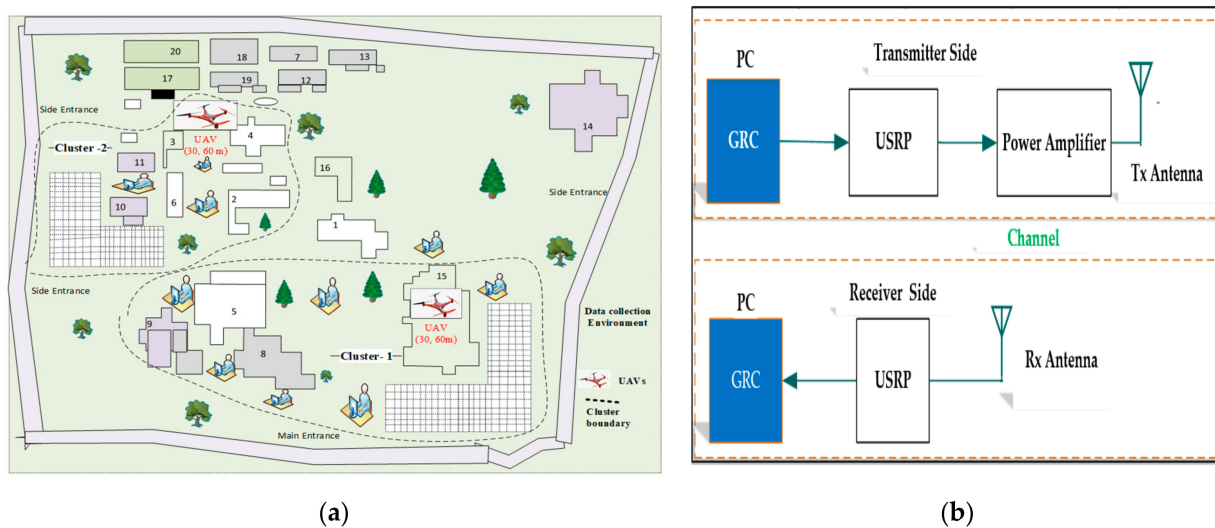


Figure 3. (a) Actual NTUT data collection layout. (b) Architecture of orthogonal frequency-division multiplexing (OFDM) signal collection strategy from UAV-base stations (Bs).

In Figure 3b for the sideways of Rx at each grid level, the machine reads and stores the OFDM signals via the Universal Software Radio Peripheral (USRP) unit. The initial code and design of the GNU Radio Companion (GRC) was compiled with C language. Then, the Ubuntu platform was used for the implementation of a software-oriented signal. The average bandwidth for UAV-BS is 3 MHz. Once the altitude of the UAV-BS increased and the LOS remained constant, the frequency of the signal quality obtained decreased at each grid level. Nevertheless, if there are buildings and trees, the UAV-BS elevation could be high. Therefore, Tx could transfer better-improved signal values than at low-level elevation.

4.2. Data Collection Procedures

For the management of resources, we conducted a real experiment involving data measurement at the NTUT, Taiwan. To collect our data, UAVs were used as Tx. The data measurements were performed at both Tx-Rx sides as a base station and user equipment (UE) features. Following Reference [5], the location pair of horizontal and vertical (elevation) angles were enabled to estimate the direction of all resources. In each cluster, square grid areas were used for OFDM signal collection. We used a variety of devices to represent ecological characteristics because it can be used to reduce the signal variations and processing time of the dataset. Hence, smartphones such as Samsung, Huawei, and Apple were used. At each grid level, the 35 OFDM signal was composed within a 10 second interval from one reference point to the next. Three separate days were used for data collection. Hence, to match a real situation with the environment, we measured UAV elevations at 30 m and 60 m with the distributed mobile users. The dipole antennas on the UAV and a USRP unit were used. A power amplifier and processor were also used to measure the actual OFDM signal for each Tx to Rx. The USRP device was used on the Tx side to shape the software-defined radio (SDR) scheme. Additionally, the output power of the Tx antenna was set to 29.5 dBm. Table 1 illustrates the sample parameter values with the UAV-BS. The remaining parameters are as follows. For 30 m and 60 m altitudes, the

cluster index is 1 (C_1) and cluster 2 (C_2). The Rss_1 signal values are used as ($Rss_1, Rss_2, \dots Rss_n$), signal-noise ratio (SNR) values are used as ($SNR_1, SNR_2, \dots SNR_n$) for training. The throughput values are expressed by the values of $T_1, T_2 \dots T_n$ for UAV-BS at 30 m and 60 m and each cluster index of 1 and 2, respectively. The power transmission values are expressed as $Pt_1, Pt_2 \dots Pt_n$ for UAV-BS for both 30 m and 60 m.

Table 1. Description of record values for training datasets.

UAV-BS (30 m)						UAV-BS (60 m)				
Cluster Index	Alt. (m)	Rss Values	SNR Values	Throughput Value	Power Value	Alt. (m)	Rss Values	SNR Values	Throughput Value	Power Value
C_1	30	Rss_1	SNR_1	T_1	Pt_1	60	Rss_1	SNR_1	T_1	Pt_1
		Rss_2	SNR_2	T_2	Pt_2		Rss_2	SNR_2	T_2	Pt_2
		\dots	\dots	\dots	\dots		\dots	\dots	\dots	\dots
		Rss_n	SNR_n	T_n	Pt_n		Rss_n	SNR_n	T_n	Pt_n
		Rss_1	SNR_1	T_1	Pt_1		Rss_1	SNR_1	T_1	Pt_1
C_2	30	Rss_2	SNR_2	T_2	Pt_2	60	Rss_2	SNR_2	T_2	Pt_2
		\dots	\dots	\dots	\dots		\dots	\dots	\dots	\dots
		Rss_n	SNR_n	T_n	Pt_n		Rss_n	SNR_n	T_n	Pt_n

Table 2 contains the parameters for data collection requirements. Specifically, we applied a clustering approach (i.e., K-means) to cluster the dataset. Then, cluster 1 and cluster 2 are obtained based on their LOS/NLOS view, signal strength values, and fluctuation features.

Table 2. List of notations.

Parameter	Description	Parameter	Description
P_L	Path loss	T_{eu}	Effective user throughput
T_x	Transmitter	T_s	Time slot
M	UAV	T_s/s	Produced signal values per time slot
U	Users equipment's	G_{Tx}, G_{Rx}	T_x and R_x antenna gains
R_x	Receiver	λ	Wavelength of the radio rate
D_{um}	Altitude from M and U (30, 60 m)	Rss_n	Received signal strength
f_c	Carrier frequency (900 MHz)	SNR	Signal noise ratio
PL^{mu}	Path loss between M and U	R_1, R_2, R_3	Resources 1, resource 2, resource 3
S^{mu}	Small-scale fading coefficient between M to U	S_t	State value at a time
$C(D_{um})$	Channel gain from M to U	Q_t	Target value at a time
Pt^1	Power transmission one	A_t	Action value at a time
$C^{D_{um}1}$	Channel gain from UAV to user 1	S'_t	Next state value
N_0	Noise power	R_t	Reward value a time
$D_{um}n$	Altitude from M to user n	γ	Represent to control the changing situations
Phy_u	Physical capacity data rate	π_t^*	Value of optimal policy
$PhyD_{um}$	Physical capacity data rate from M to U	A_t'	Next action value
T_U	User throughput		

Figure 4 illustrates the methods of data collection, preprocessing, training, and regression phases after the dataset are obtained. Initially, users are randomly distributed to the environment (Stage 1). Stage 2: the clustering phase algorithm is applied to cluster based on user locations, signal variations, and environmental features. Stage 3: preprocessing of the dataset is done. After composing the required dataset, the training process is carried out at Stage 4. Bandwidth, throughput, power gain, and SINR values are selected as target parameters for the RM optimization model. Then, a round robin scheduling-algorithm is applied for sorting the resource request queue. The DRL is precisely trained based on user service requests. In stage 5, testing and regression tasks are carried out with a

multi-agent RM approach, and the resources are allocated. Initially, users generate services from the sources. Then, users acquire their resource interest. The algorithm determines the availability of resources in parallel with the resource request. The classifier method aims to sort resource requests based on their preferences for assigning a ranking order. A comparison of requests is made for the execution phase. In the end, scheduling is done for the available resources. The architecture is cyclical and performance is evaluated until the optimal result is obtained.

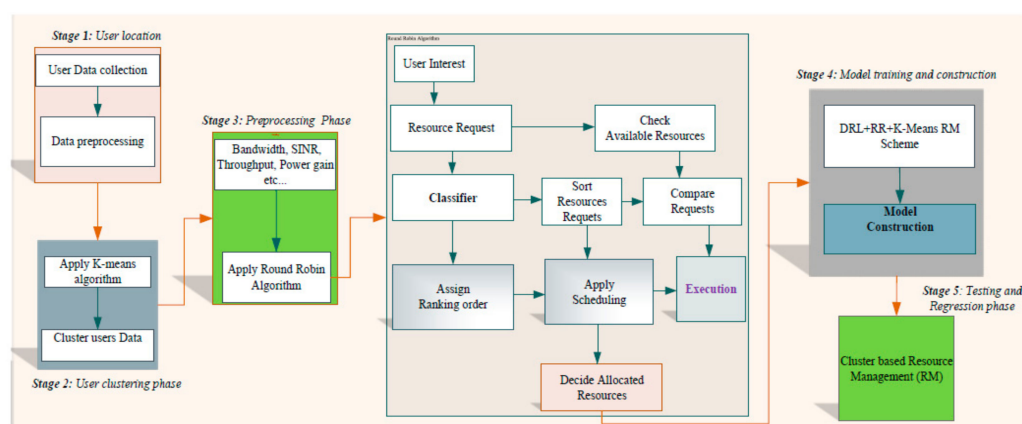


Figure 4. The flow chart of clustering, training and scheduling phases: DRL; deep reinforcement learning.

5. Proposed Method Description

This section contains the general architecture of DRL for our system, the round robin scheduling-algorithm for resource management, multi-agent DRL for the joint resource management problem, the detailed architecture of the proposed system, and evaluation metrics.

5.1. Round-Robin Scheduling Algorithm for Resource Management

According to Reference [26], round robin is a scheduling algorithm in which each resource can be assigned with a duration of time and iteration. The algorithm mainly focuses on the time slot and time-sharing-based scheduling, and is applied to ensure that resources are allocated fairly at each time slot. If the RM processes are not assigned or completed at a given time, the allocation queue comes after other resources have arrived, which makes scheduling fair [19]. Therefore, the round robin scheduling-algorithm has various advantages, such as being easy to implement, cyclical, and starvation not occurring. For different resources, it supports first come and first serve to schedule, and gives equal access to all resources to be allocated. When a new resource request comes in, it is added to the end of the queue and is ready to be managed and allocated. Therefore, each resource has a chance of being rescheduled for allocation with a particular time slot. Generally, the following steps are applied for our proposed system to compute the resource management requests:

- (1) Decide the resource that comes first and then start to allocate the resources as a time slot only.
- (2) Check the other resource requests. If there is a resource request that is available in a one-time slot while another resource request is being filled, the incoming resources are put on a waiting list as a ready queue.
- (3) After the time slot has gone, check for any more resource requests in the queue. If the existing RM process is not finished, add the existing request to the end of the queue.
- (4) Take the first request from the waiting ready queue and start to allocate it (same rules),
- (5) Steps (2)–(4) can be repeated.
- (6) If the resource request is finished and none are waiting in the queue, then the assigning work is done.

For instance, Table 3 illustrates the resource management process with the round robin scheduling-algorithm. It contains a list of resources (resource 1 (R1), resource 2 (R2), and resource 3 (R3)), duration, the order of the queue, and arrival time slots. Therefore, R1, R2, and R3 are given a waiting time slot of four (4), six (6), and six (6), respectively. Hence, the average waiting time (AWT) of all resource requests is $(4 + 6 + 6)/3 = 5.33$. Lastly, three-time slot considerations are crucial for applying RR as an RM issue.

Table 3. Illustration of record of resources with a round robin scheduling-algorithm.

Resources	Duration	Queue	Arrival Time
R1	3	0	0
R2	4	1	0
R3	3	2	0

Then, we consider the time slot value is one (1):

R1	R2	R3	R1	R2	R3	R1	R2	R3	R2
0									10

- Completion Time: the time slot at which the resource request allocation assignment ends.
- Turnaround Time: the total time that the service request is available in the queue (i.e., turnaround time = completion time – arrival time).
- Waiting Time (WT): the total waiting duration (i.e., waiting time = turnaround time – burst time).

5.2. Multi-Agent DRL for the Joint Resource Management Problem

According to Figure 5, a multi-agent DRL is used to build a joint strategy for resource management. It is because the cumulative reward values of one user equipment are certainly influenced by other user equipment's actions in the resource management process. Hence, the initial state and UAV to user actions affect the value of the reward.

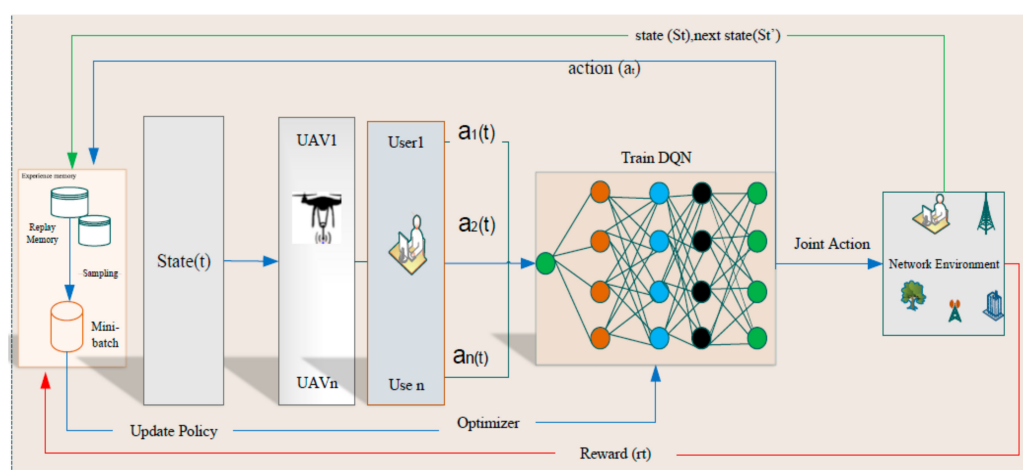


Figure 5. An illustration of multi-agent framework for proposed DRL-based RM algorithm.

Therefore, Equations (9)–(19) illustrate the relationship between the state, action, and reward for a multi-agent DRL-based scheme.

- (1) State: The A2G channel used by IoT users, the consumption of power rate, and the effects of interference are considered as a state. Then, the values of each state (S_t) with the state-action pairs at a time slot as $Q_t(S_t, a_t)$ for the next state values as $Q_t(S'_t, a_t)$. Thus, for Equation (9), S_t and S'_t are the values of a state and the next state,

respectively. Moreover, r_t , is the value of the reward; a_t is the value of action at a time slot; γ is used to represent and control the changing situations. The value of π_t^* represents the policy. Next, the agent will have access to the next state of S'_t .

$$Value(S_t) = \max_{a_t} (r_t(S_t, a_t) + \gamma \sum S'_t P(S_t, a_t, S'_t)), \quad (9)$$

$$Q_t(S_t, a_t) = r_t(S_t, a_t) + \sum S'_t \pi_t^*(S_t, a_t, S'_t) Value(S'_t) \quad (10)$$

Then, the expected values of the next state, as in Equation (10), where Q_t is the target value; (S'_t) used for the next state and action as in $Q_t(S'_t, a'_t)$. Then, the time slot calculation for the state to action communicates, as in Equations (11) and (12). Then, the time slot (T_s) is calculated as in Equation (13):

$$Q_t(S_t, a_t) = r_t(S_t, a_t) + \sum S'_t \pi_t^*(S_t, a_t, S'_t) \max_{a'_t} Q_t(S'_t, a'_t), \quad (11)$$

$$Q_t(S_t, a_t) = r_t(S_t, a_t) + \gamma \max_{a'_t} Q_t(S'_t, a'_t), \quad (12)$$

$$T_s(a_t, S_t) = r_t(S_t, a_t) + \gamma \max_{a'_t} Q_t(S'_t, a'_t) - Q_t(S_t, a_t), \quad (13)$$

- (2) Joint action selection strategy: The allocation of resources such as channel allocation of bandwidth, throughput, and power is the action taken by the agent. This is used to equate action1 with next action vs. action1 to target1, and action n to target n. Where, to control the randomness of the environment, add the time slot as in Equation (14). Based on Equation (13) of $Q_t - 1(S_t, a_t)$ and the time slot of $Q_t(S_t, a_t)$. Then, as in Equation (15), this shows how our target (Q_t) values are learned from the state and updated over time with $T_s(a_t, S_t)$ as in Equation (16):

$$Q_t(S_t, a_t) = Q_t - 1(S_t, a_t) + a_t T_{st}(a_t, S_t) \quad (14)$$

$$Q_t(S_t, a_t) = Q_t - 1(S_t, a_t) + a_t (r_t(S_t, a_t) + \gamma \max_{a'_t} Q_t(S'_t, a'_t) - Q_t - 1(S_t, a_t)) \quad (15)$$

$$T_s(a_t, S_t) = r_t(S_t, a_t) + \gamma \max_{a'_t} Q_t(S'_t, a'_t) - Q_t - 1(S_t, a_t) \quad (16)$$

Next, the loss function is computed as in Equation (17), with the sum of the Q-values and their target differences as follows:

$$Loss = \sum (Q_t - Target - Q_t)^2 \quad (17)$$

- (3) Reward: users need to achieve QoS constraints such as throughput, power, and bandwidth called rewards. However, the SINR values must be higher than the SINR threshold. Therefore, the reward is the effective management of resources; otherwise, the reward is zero. As UEs takes the action of $Q_t(S_t, a_t)$ by observing $Value(S_t)$, it obtains the immediate $Reward(r_t)$. After an action is taken, reward the agent based on the result obtained. Then, the value of the new state is computed, as in Equation (18):

$$Reward(S_t) + \gamma \max_{a'_t} Q_t(S_t, a'_t), \quad Reward = r_t(S_t, a_t) + \gamma \max_{a'_t} Q_t(S'_t, a'_t) \quad (18)$$

- (4) Replay memory: It used to save the evaluation of the cost of past experiences, i.e., past state-action pairs in our system. The combination of using the batch method and replay memory improves the convergence of DRL [28]. Then, for each episode, take a batch of samples from the past experiences. Then, find the gradient of the weights and store it into replay memory. Finally, train the weights of the deep neural network to estimate the loss function.
- (5) Policy: updating the value function, the agent also needs to sample and learn the environment by performing some non-optimal policy (i.e., the optimal policy π_t^* : $S_t \rightarrow a_t$ is obtained to maximize long-term reward). Then, each UE learns the optimal policy $a_t^* = \pi_t^*(S_t) \in a_t$ based on the state space through the A2G access link from the associated UAV. Therefore, all the agents need to obtain the optimal policy.

5.3. Proposed Method Architecture

In the working environment, each user observes and submits requests for services. Additionally, users involve activities in the state; UAV takes action to design and learn suitable policies for the next state. The environmental capability of individual users has been observed. After the DRL agent builds its state based on previous actions and local observations, then determine the state of the environment. Local observations are used to provide and enhance the observation of the global state in the environment [26]. Figure 6 shows the structure of our proposed system. The figure considers the principle of DRL for the RM parameters of power (Pr), throughput (Tu), and SINR. Therefore, we need to correlate the state, environment, action, reward, policy, and training parameters. The format of the figure is as follows. Users and UAVs are regarded as agents. The data collection area of NTUT (500×700 m) is the environment. The transmission of the A2G channel allowed for cooperation between users and UAVs, the resources (bandwidth, power, throughput, and SINR) are allocated. Then, each participating agent attempts to assign power with the throughput and management of SINR as an action. When resources are efficiently managed, lower than the threshold, it is a reward. Thus, when the threshold matches the situation, the reward value is effective; otherwise, the reward is zero. The action selection strategy is used to balance exploration and exploitation [13]. Hence, there may be actions conducted randomly for searching the effects of unknown actions, so it is easier to devise policies.

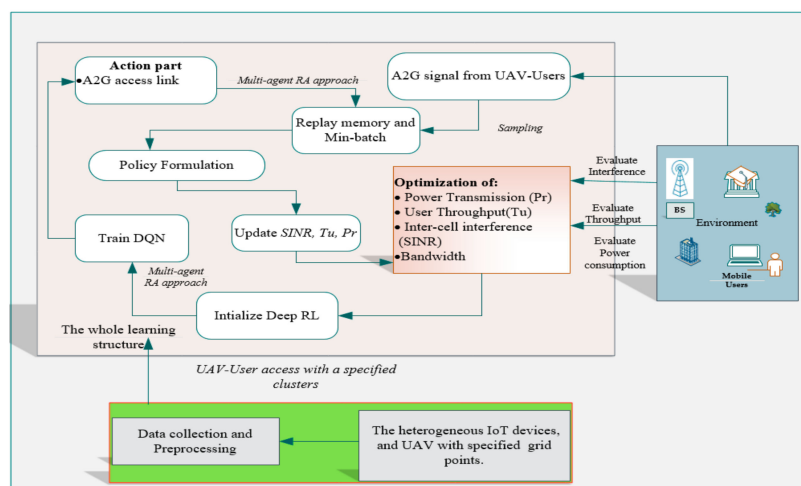


Figure 6. Proposed method architecture. A2G: air-to-ground; SINR: signal-to-interference-plus-noise ratio.

By compiling all datasets, preprocessing was done. Initially, the DRL algorithm is initialized and trained. Then, the next action component is evaluated based on A2G access links with the multi-agent DRL. Mainly, information is transmitted from the A2G channel for IoT users. Then, the system took samples for policy formulation after saving to a mini-batch to a replay memory. Then, evaluate the capacity of SINR, power and throughput, and average values. Based on the generated results, a new policy is formulated for the next active state. Then, the policy is updated for the optimization of RM. These proposed rotations will have to be conducted until we find the optimal solution. Then, the reward offered to the agent included the average reward values of every agent. If the observable values changed and the value of each reward is more than the threshold, the event is activated. Then, the action and policy are updated by an agent unless the actions are ended.

Generally, when an agent exceeds a given threshold, it learns from the environment for the new solution. Past experiences are used for learning through batch replay memory. Then, based on the previous experiences, the agent randomly picks from the uniformly distributed samples in the batch.

5.4. Evaluation Metrics'

For the accurate evaluation of our proposed system, the evaluation metrics, such as root mean square error (RMSE), precision, and recall are used. The precision and recall can both indicate the accuracy of the model. Consider the proportion of the sum of significant results properly categorized by our proposed model, to evaluate the proportion of significant outcomes, and computed as follows:

$$\text{RMSE} = \sqrt{\sum_{i=1}^n \frac{(y_i - \bar{y})^2}{n}}, \text{ Precision} = \frac{TP}{AV} \text{ or } \frac{TP}{TP + FP}, \text{ Recall} = \frac{TP}{PR} \text{ or } \frac{TP}{TP + FN}, \quad (19)$$

where n is the total forecasted values; y_i is the real value; and \bar{y} is the expected value in terms of the UAV to user altitude. Then, T_P refers to true positives, A_V refers to actual values, and P_R , F_N , and F_P are the predicted results, false negatives, and positives, respectively.

6. Experimental Results and Discussions

The experimental outcomes are illustrated in this section. The real data collected from the outdoor environment were used for the evaluation as well as analysis of the results. Preprocessing was performed to label each signal record and minimize the number of irrelevant features. Then, an appropriate and structured training dataset for machine training formats was built. The performance of the proposed method is analyzed through different measurement metrics. Then, different parameters, such as the number of epochs, hidden layers and units, types of optimization, and the activation function were adjusted. Our proposed approach was trained using adam optimizer because it is computationally efficient, requires less memory, and minimize noise. Additionally, specific activation functions such as sigmoid, reLU, tanH, and softmax were used to compress the output of the system. As a result, the optimum values were obtained with a batch size of 500, 200 and 250 epochs/iterations, four hidden layers, and 824 hidden units. The classification and testing schemes were executed through the scenarios, which prove the effectiveness of the proposed approach. Accordingly, the comparison of our system performances with other related algorithms based on clusters 1 and 2 scenarios with UAV altitudes of 30 and 60 m were conducted. Table 4 shows the hardware and software experimental setups for implementing the proposed system. The Keras library of a TensorFlow 3.7 backend with a combination of a variety of deep learning frameworks has been chosen. This is because it is easy to use, has better visualization and has optimized numerical computations.

Table 4. Hardware and software resources used.

Category	Tools
Programming language	Python 3.7
Library	Keras libraryTensorFlow
CPU	Intel(R) Core(TM) i7-7700 CPU@ 3.60GHz
System Type	64-bit operating system, ×64-based processor
GPU	NVIDIA 1080TI
RAM	16 GB

Figure 7a shows the DRL-based classification evaluation of the real distribution OFDM signal values. Then, it is illustrated through the training and testing evaluation. The OFDM_data values are represented in blue color and range from -40 to -85 ; the training_set and testing_set values are represented in yellow and green colors, respectively. Figure 7b shows the clustering of signal distribution. It shows the real distribution of OFDM signal values in terms of user density. The green squares indicate the signal distribution of cluster 1. The black squares show cluster 2's plotting results. The plotting is performed through the consideration of signal distribution ranges and the LOS/NLOS estimation.

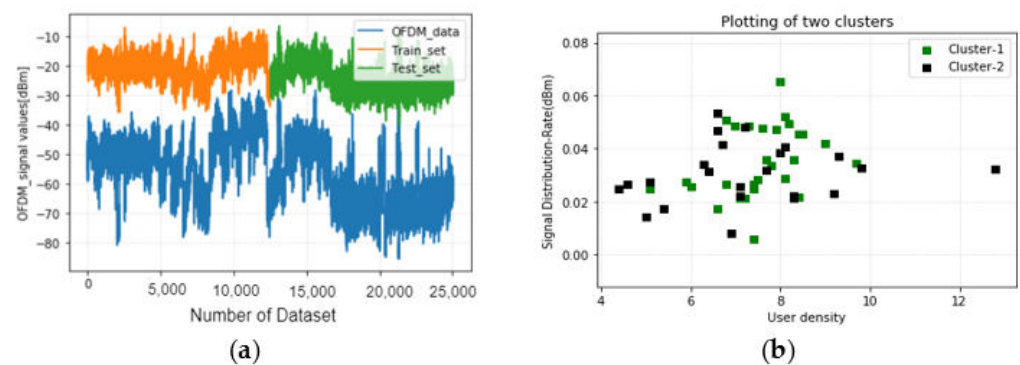


Figure 7. (a) Training vs. testing evaluation based on deep reinforcement learning (DRL); (b) Illustration of clustering classes.

Table 5 shows the performance evaluation of single-agent Q-learning and DRL approaches for clusters 1 and 2. As measurement metrics, accuracy, precision, and recall are used. The evaluation is concerned with the average allocation of throughput, interference management, and power consumption issues. In the Q-learning method, 87.03%, 75.00%, and 80% and 85.00%, 80.01%, and 85.02% are the values for accuracy, precision, and recall cluster 1 and cluster 2, respectively. In the DRL method, 94.06%, 88.05%, and 90.01% and 91.2%, 92.00%, and 93.04% are the values for accuracy, precision, and recall in cluster 1 and cluster 2, respectively. Therefore, the evaluation result shows that the performance accuracy is better in DRL evaluation for both clusters for multi-action-based RM optimization in a real and dynamic environment. Thus, multi-agent DRL is better in retrieving relevant features than single-agent Q-learning methods in complex and dynamic environments.

Table 5. Performance evaluation.

Evaluation Metrics	Q-Learning		DRL	
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
Accuracy (%)	87.03	85.00	94.06	91.2
Precision (%)	75.00	80.01	88.05	92.00
Recall (%)	80.00	85.02	93.04	90.01

Using pie charts, Figure 8 shows each resource management scheme and calculates the QoS satisfaction ratio. The figure provides a detailed performance comparison among the resource management parameters, where the results for the DRL method are obtained with different learning updates. The percentage of resource requests for cluster 1 (urban) and cluster 2 (sub-urban) are illustrated in Figure 8a, b. Figure 8a accounts for 52.9% of throughput estimation, 37.3% of power consumption estimation, and 9.8% of SINR estimation. Figure 8b accounts for 52.9% of throughput estimation, 43.1% of power consumption estimation, and 3.9% of SINR estimation. This implies that more resource values are shared between users with the minimum values of power consumption and SINR. The resource management capability of our proposed system with a UAV altitude of 30 m and 60 m, respectively, is illustrated in Figure 8c,d. When the A2G access link transmission involves 30 m and 60 m altitudes, the management of resources leads to a 31.4% and 90.2% throughput estimation user satisfaction ratio, 58.8% and 5.9% power consumption estimation ratio, and 9.8% and 3.9% SINR estimation ratio. Therefore, DRL-based resource management is best when the altitude of the UAV is larger (i.e., 60 m) because its value is automatically changed and it has better performance in terms of user throughput values and lower values of power consumption and SINR. However, Figure 8e shows the management of resource estimations without applying the scheduling algorithm (i.e. round-robin). Even if the result is not bad, it is lower compared to others. The resource management of user throughput ratio, power consumption estimation ratio, and SINR estimation ratio is

7.8%, 82.4%, and 9.8%, respectively. Therefore, the integration of the DRL system with the round robin scheduling algorithm performs better for the UAV-based IoT user resource management scheme.

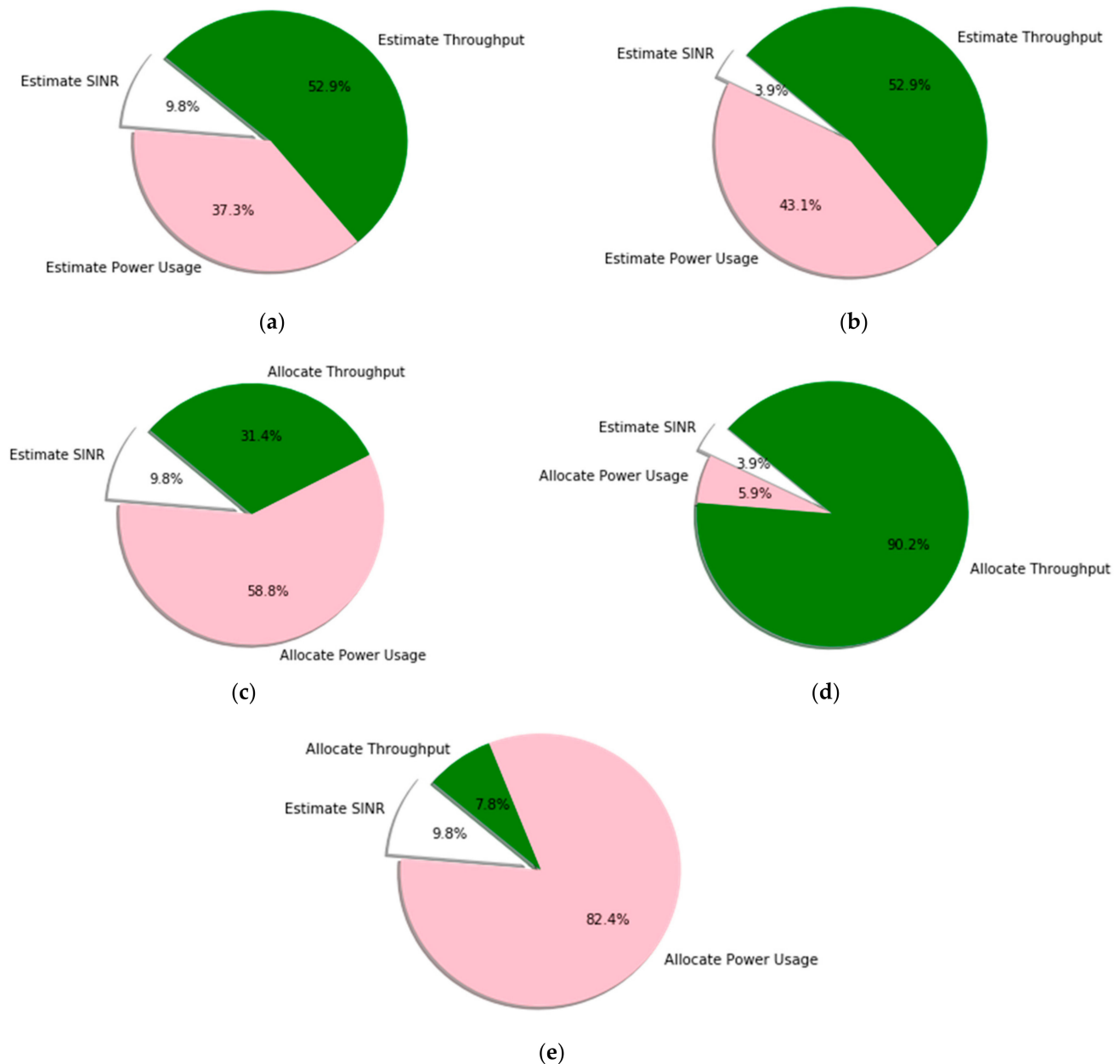


Figure 8. The average resource management of the target clusters for different schemes in different learning processes: (a) service request at cluster 1, (b) service request at cluster 2, (c) resource management at 30 m UAV altitude, (d) resource management at 60 m UAV altitude, and (e) average resource management without applying round robin.

Figure 9a,b evaluates the DRL-based average A2G wireless latency values and the data access rate of A2G IoT user equipment. In the figures, the measurement effects enable us to precisely control the user throughput variance at each reference point. As a result, taking into account UAV heights, the proposed system will accommodate user throughput issues. It handles multiple IoT users with varying UAV altitudes and ranges. Moreover, even when users are shifted, the proposed DRL model can accurately measure user throughput and power consumption rates.

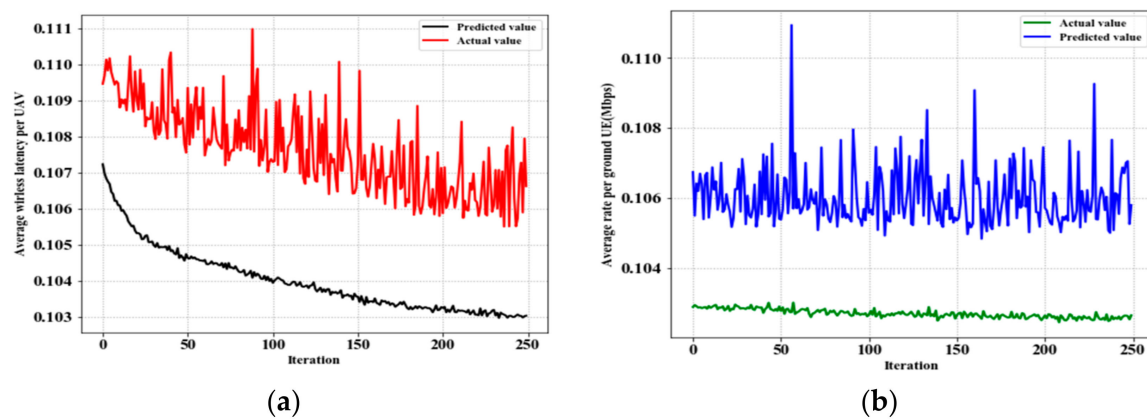


Figure 9. Performance analysis of average throughput and power rates: (a) average throughput delay from the UAV and (b) data access rate per ground user equipment as compared to various iterations.

Under different scenarios, Table 6 shows the user throughput performance, average power consumption, and average SINR distribution with distinct user density. The user throughput value accuracy assessment is estimated to be 0.96, 0.967, 0.967, 0.97, 0.97, and 0.97, respectively, when the user density is 1, 3, 6, 11, 21, and 26. This illustrates that, when user density is increased, it also improves the value of user throughput performance. This shows that the algorithm can manage a large number of IoT users in a dynamic environment. The energy of average consumption is estimated as 0.96, 0.965, 0.967, 0.967, 0.967, and 0.96, when the user density becomes 1, 3, 6, 11, 21, and 26, respectively. From the Tx to the Rx is diminished when the number of IoT users is increased, which means a balance of power usage is maintained. The SINR and user density are also calculated to be 0.96, 0.97, 0.75, 0.96, 0.966, and 0.966 where the user density is 1, 3, 6, 11, 21, and 26, respectively. This demonstrates that, when the number of IoT users is increased, the SINR from a transmitter to a receiver is significantly decreased. The proposed algorithm can handle the interference between a user transmitter or an A2G access channel, even though the result is better as the number of IoT users increase. The performance measurement is then carried out using average user throughput data, average power usage, and average SINR results from the target values of clusters 1 and 2. Therefore, in terms of managing resources for IoT users in each cluster, the proposed framework contributes to better performance.

Table 6. The average accuracy of the target clusters and the user density in different learning processes.

User Density	Performance Outcomes					
	1	3	6	11	21	26
Average throughput rate [Mbps] (%)	0.96	0.967	0.967	0.97	0.97	0.97
Average energy consumption rate [dBm] (%)	0.96	0.965	0.967	0.967	0.967	0.96
Average SINR level [dBm] (%)	0.96	0.97	0.75	0.966	0.966	0.966

Figure 10 shows the accuracy of the average values for user throughput in the target area based on the DRL method of iterations. As presented in the figure, the real and forecasted throughput values converge rapidly at a value of 0.015 after 20 iterations. Subsequently, the actual values are closely aligned with the expected values. It is feasible that the network cannot be over adapted to the training data.

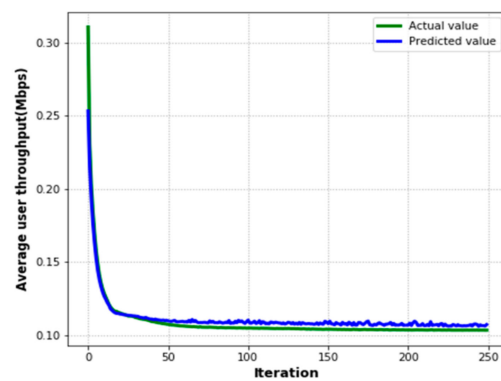


Figure 10. Actual and predicted value evaluation of throughput distributions.

Table 7 contains the sample performance comparison of our system with previous related works. Previous studies [22,29] applied DRL and Q-learning+D3QN algorithms, respectively. At the same time, our work used DRL+K-means and integrated the scheduling algorithm of round robin. The accuracy performance is 87%, 81.3%, and 94% for the specified related works and our proposed method estimation, respectively. The RMSE estimation evaluation indicates 5.2%, 7.33%, and 2.40% for the specified related works and our proposed method estimation, respectively. The testing time is 2.5%, 2.01%, and 1.05% for the specified related works and our proposed method estimation, respectively. Hence, our proposed system outperforms other related works. Compared to the performance in similar scenarios, all the previous methods (i.e., DRL, Q-learning+D3QN) achieved lower system accuracy and RMSE evaluation.

Table 7. Comparison of our proposed system with previous works.

Year	Algorithm Used	Accuracy	RMSE	Testing Time(s)
2018 [22]	DRL	87%	5.2%	2.5
2019 [29]	Q-learning + D3QN	81.3%	7.33%	2.01
Our proposed method	DRL + K-means + Round-robin	94%	2.40%	1.05

Therefore, our proposed method shows better performance in terms of accuracy and RMSE testing time evaluations. This implies that our system has good generalization ability, low computational time, and adaptability to real and dynamic IoT environments. Furthermore, this indicates that the proposed DRL+K-means+round-robin method has capability in terms of solving large-scale learning problems. Therefore, the testing time of the proposed model is smaller than the other models and this indicates that the proposed model can be trained faster.

7. Conclusions

By applying the DRL, this study focuses on UAV-based resource management on cellular and IoT networks. Initially, we started by identifying the challenges of resource management in IoT networks assisted by UAV-BSs. Then, we reviewed the traditional resource management mechanisms for IoT networks and assessed the usage of DRL techniques for resource management. Subsequently, a multi-agent DRL approach was proposed in order to obtain an efficient resource management method for UAV-assisted IoT communication systems. The resource management algorithm is used to manage the bandwidth, throughput, interference, and power usage issues. First, we looked at the actual data collection setting for joint RM issues. Then, we used a DRL for system development with K-means for clustering as well as round robin for service request queue. To improve the RM scheme, our proposed approach allows for allocating the available resources with UAV

to IoT users. Then, with the measurement of accuracy, RMSE and testing time(s), our proposed method was compared to previous works. The proposed system was found to have a 94% RM accuracy with respect to the classification scheme. It achieved RMSE and testing times(s) levels of 2.40% and 1.05s, respectively, for the selected scenarios. Additionally, our method was found to have a better evaluation of precision and recall estimation than the Q-learning approach as 88.05%, 93.04% and 92%, 90.01% for cluster 1 and 2, respectively. Thus, the result demonstrates the well-trained multi-agent DRL learned from sequential features. The proposed method is better for generating optimal policy with low computational complexity than a single-agent-based Q-learning method. Therefore, the proposed DRL model can manage the resource management scheme in dynamic IoT environments. For future work, this study will be expanded to take into account the mobility of users in dynamic environments with a focus on additional resource considerations.

Author Contributions: Conceptualization, H.-P.L., and D.-B.L.; Formal analysis, R.-T.J.; Funding acquisition, H.-P.L., and D.-B.L.; Methodology, Y.Y.M.; Project administration, H.-P.L.; Software, Y.Y.M., and G.B.T.; Supervision, R.-T.J., and D.-B.L.; Validation, G.B.T.; Writing—original draft, Y.Y.M.; Writing—review & editing, R.-T.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Ministry of Science and Technology (MOST), Taiwan, under Grants 109-2222-E-035-003-MY2.

Data Availability Statement: The data and codes presented in this study are available from the corresponding author by request.

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

References

1. Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1251–1275. [\[CrossRef\]](#)
2. Munaye, Y.Y.; Lin, H.-P.; Adege, A.B.; Tarekegn, G.B. UAV Positioning for Throughput Maximization Using Deep Learning Approaches. *Sensors* **2019**, *19*, 2775. [\[CrossRef\]](#)
3. Zhang, Z.; Yang, G.; Ma, Z.; Xiao, M.; Ding, Z.; Fan, P. Heterogeneous Ultra-Dense Networks with NOMA: System Architecture, Coordination Framework, and Performance Evaluation. *IEEE Veh. Technol. Mag.* **2018**, *13*, 110–120. [\[CrossRef\]](#)
4. Shen, K.; Yu, W. Fractional Programming for Communication Systems-Part I: Power Control and Beamforming. *IEEE Trans. Signal Process.* **2018**, *66*, 2616–2630. [\[CrossRef\]](#)
5. Munaye, Y.Y.; Adege, A.B.; Tarekegn, G.B.; Li, Y.; Lin, H.; Jeng, S. Deep Learning-Based Throughput Estimation for UAV-Assisted Network. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
6. Sun, H.; Chen, X.; Shi, Q.; Hong, M.; Fu, X.; Sidiropoulos, N.D. Learning to Optimize: Training Deep Neural Networks for Interference Management. *IEEE Trans. Signal Process.* **2018**, *66*, 5438–5453. [\[CrossRef\]](#)
7. Wang, H.; Wang, J.; Ding, G.; Wang, L.; Tsiftsis, T.A.; Sharma, P.K. Resource Allocation for Energy Harvesting-Powered D2D Communication Underlying UAV-Assisted Networks. *IEEE Trans. Green Commun. Netw.* **2018**, *2*, 14–24. [\[CrossRef\]](#)
8. Tarekegn, G.B.; Juang, R.-T.; Lin, H.-P.; Adege, A.B.; Munaye, Y.Y. DFOPS: Deep Learning-Based Fingerprinting Outdoor Positioning Scheme in Hybrid Networks. *IEEE Internet Things J.* **2020**, *8*, 3717–3729. [\[CrossRef\]](#)
9. Qin, Z.; Yue, X.; Liu, Y.; Ding, Z.; Nallanathan, A. User Association and Resource Allocation in Unified NOMA Enabled Heterogeneous Ultra-Dense Networks. *IEEE Commun. Mag.* **2018**, *56*, 86–96. [\[CrossRef\]](#)
10. Zhang, H.; Feng, M.; Long, K.; Karagiannidis, G.K.; Nallanathan, A. Artificial Intelligence-Based Resource Allocation: Applications in Ultra Dense Networks. *IEEE Veh. Technol. Mag.* **2019**, *14*, 56–63. [\[CrossRef\]](#)
11. Zeng, Y.; Zhang, R. Energy-Efficient UAV Communication with Trajectory Optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [\[CrossRef\]](#)
12. Liu, S.; Wei, Z.; Guo, Z.; Yuan, X.; Feng, Z. Performance Analysis of UAVs Assisted Data Collection in Wireless Sensor Network. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–5.
13. Liang, L.; Ye, H.; Yu, G.; Li, G.Y. Deep-Learning-Based Wireless Resource Allocation with Application to Vehicular Networks. *Proc. IEEE* **2020**, *108*, 341–356. [\[CrossRef\]](#)
14. Markakis, E.K.; Karras, K.; Sideris, A.; Alexiou, G.; Pallis, E. Computing, caching, and communication at the edge: The cornerstone for building a versatile 5G ecosystem. *IEEE Commun. Mag.* **2017**, *55*, 152–157. [\[CrossRef\]](#)
15. Karras, K.; Pallis, E.; Mastorakis, G.; Nikoloudakis, Y.; Batalla, J.M.; Mavromoustakis, C.X.; Markakis, E. A hardware acceleration platform for AI-based inference at the edge. *Circuits Syst. Signal Process.* **2020**, *39*, 1059–1070. [\[CrossRef\]](#)

16. Xu, C.; Jiang, S.; Luo, G.; Sun, G.; An, N.; Huang, G.; Liu, X. The Case for FPGA-based Edge Computing. *IEEE Trans. Mob. Comput.* **2020**, *1*. [\[CrossRef\]](#)
17. Haghi Kashani, M.; Rahmani, A.M.; Jafari Navimipour, N. Quality of service-aware approaches in fog computing. *Int. J. Commun. Syst.* **2020**, *33*, e4340. [\[CrossRef\]](#)
18. Sun, Y.; Peng, M.; Mao, S. Deep reinforcement learning-based mode selection and resource management for green fog radio access networks. *IEEE Internet Things J.* **2019**, *6*, 1960–1971. [\[CrossRef\]](#)
19. Ahmed, K.I.; Hossain, E. A Deep Q-Learning Method for Downlink Power Allocation in Multi-Cell Networks. *arXiv* **2019**, arXiv:1904.13032.
20. Zhang, Q.; Mozaffari, M.; Saad, W.; Bennis, M.; Debbah, M. Machine Learning for Predictive On-Demand Deployment of UAVs for Wireless Communications. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 10–12 December 2018; pp. 1–6.
21. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [\[CrossRef\]](#)
22. Li, R.; Zhao, Z.; Sun, Q.; Chih-Lin, I.; Yang, C.; Chen, X.; Zhang, H. Deep Reinforcement Learning for Resource Management in Network Slicing. *IEEE Access.* **2018**, *6*, 74429–74441. [\[CrossRef\]](#)
23. Liu, B.; Xu, H.; Zhou, X. Resource Allocation in Unmanned Aerial Vehicle (UAV)-Assisted Wireless-Powered Internet of Things. *Sensors* **2019**, *19*, 1908. [\[CrossRef\]](#)
24. Li, Y.; Gao, Z.; Huang, L.; Du, X.; Guizani, M. Resource management for future mobile networks: Architecture and technologies. *Comput. Netw.* **2017**, *129*, 392–398. [\[CrossRef\]](#)
25. Yang, R.; Ouyang, X.; Chen, Y.; Townend, P.; Xu, J. Intelligent Resource Scheduling at Scale: A Machine Learning Perspective. In Proceedings of the 12th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2018 and 9th International Workshop on Joint Cloud Computing (JCC), Oxford, UK, 23–26 August 2018; pp. 132–141.
26. Calabrese, F.D.; Wang, L.; Ghadimi, E.; Peters, G.; Hanzo, L.; Soldati, P. Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges. *IEEE Commun. Mag.* **2018**, *56*, 138–145. [\[CrossRef\]](#)
27. Vamvakas, P.; Tsiropoulou, E.E.; Vomvas, M.; Papavassiliou, S. Adaptive power management in wireless powered communication networks: A user-centric approach. In Proceedings of the 2017 IEEE 38th Sarnoff Symposium, Newark, NJ, USA, 18–20 September 2017; pp. 1–6.
28. Wang, S.; Liu, H.; Gomes, P.H.; Krishnamachari, B. Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 257–265. [\[CrossRef\]](#)
29. Zhao, N.; Liang, Y.C.; Niyato, D.; Pei, Y.; Wu, M.; Jiang, Y. Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 5141–5152. [\[CrossRef\]](#)
30. Ghadimi, E.; Calabrese, F.D.; Peters, G.; Soldati, P. A reinforcement learning approach to power control and rate adaptation in cellular networks. In Proceedings of the 2017 IEEE International Conference on Communications, Paris, France, 21–25 May 2017; pp. 1–7.
31. Munaye, Y.Y.; Juang, R.T.; Lin, H.P.; Tarekegn, G.B. Hybrid deep learning-based throughput analysis for UAV-assisted cellular networks. *IET Commun.* **2021**, *14*, 1751–8628.
32. Munaye, Y.Y.; Juang, R.-T.; Lin, H.-P.; Tarekegn, G.B. Resource Allocation for Multi-UAV Assisted IoT Networks: A Deep Reinforcement Learning Approach. In Proceedings of the 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), Taipei, Taiwan, 3–5 December 2020; pp. 15–22.
33. Lee, K.; Kim, J.; Kim, J.; Hur, K.; Kim, H. CNN and GRU Combination Scheme for Bearing Anomaly Detection in Rotating Machinery Health Monitoring. In Proceedings of the 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), Jeju Island, Korea, 23–27 July 2018; pp. 102–105.
34. Ye, H.; Li, G.Y.; Juang, B.H.F. Deep Reinforcement Learning Based Resource Allocation for V2V Communications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3163–3173. [\[CrossRef\]](#)
35. He, Y.; Zhao, N.; Yin, H. Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 44–55. [\[CrossRef\]](#)
36. Tan, L.T.; Hu, R.Q. Mobility-aware edge caching and computing in-vehicle networks: A deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10190–10203. [\[CrossRef\]](#)
37. Du, X.; Zhang, H.; Nguyen, H.V.; Han, Z. Stacked LSTM Deep Learning Model for Traffic Prediction in Vehicle-to-Vehicle Communication. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017; pp. 1–5.
38. Baker, S.B.; Xiang, W.; Atkinson, I. Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access.* **2017**, *5*, 26521–26544. [\[CrossRef\]](#)