applied sciences

MDPI

Article

Adaptive Human–Machine Evaluation Framework Using Stochastic Gradient Descent-Based Reinforcement Learning for Dynamic Competing Network

Jinbae Kim and Hyunsoo Lee *

School of Industrial Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea; dbk0508@kumoh.ac.kr

* Correspondence: hsl@kumoh.ac.kr; Tel.: +82-54-478-7661

Received: 9 March 2020; Accepted: 4 April 2020; Published: 8 April 2020

Abstract: Complex problems require considerable work, extensive computation, and the development of effective solution methods. Recently, physical hardware- and software-based technologies have been utilized to support problem solving with computers. However, problem solving often involves human expertise and guidance. In these cases, accurate human evaluations and diagnoses must be communicated to the system, which should be done using a series of real numbers. In previous studies, only binary numbers have been used for this purpose. Hence, to achieve this objective, this paper proposes a new method of learning complex network topologies that coexist and compete in the same environment and interfere with the learning objectives of the others. Considering the special problem of reinforcement learning in an environment in which multiple network topologies coexist, we propose a policy that properly computes and updates the rewards derived from quantitative human evaluation and computes together with the rewards of the system. The rewards derived from the quantitative human evaluation are designed to be updated quickly and easily in an adaptive manner. Our new framework was applied to a basketball game for validation and demonstrated greater effectiveness than the existing methods.

Keywords: adaptive human evaluation; dynamic competing network; reinforcement learning; stochastic gradient descent

1. Introduction

Artificial intelligence (AI) technologies are developing with a focus on designing systems for efficient learning, effective solution of complex problems, and rapid large-scale computation. Reinforcement learning (RL) takes the form of learning by rewarding the changing state from the action of the learning object in the defined system environment [1]. Problem solving approaches that involve RL require advanced methods due to the system complexity, as well as additional steps such as pre-learning or preprocessing. Therefore, to solve complex and difficult RL problems effectively, a strategic policy is used to update the system reward by obtaining feedback through human intervention [2,3]. Humans with expert knowledge of the problem to be solved can respond intuitively, accurately diagnose the system state, and quickly determine the required action. Therefore, the learning object can be clearly defined by utilizing the fact that it is similar to a human being. However, the accuracy of human evaluation has been reduced by designing such evaluations using binary numbers in previous studies focused on feedback by learning through human intervention.

In this study, to address these shortcomings, algorithms were designed to ensure accurate and clear learning through quantitative evaluation in the form of real numbers. In addition, the stochastic gradient descent (SGD) algorithm was used to overcome the disadvantage of slow learning when human intervention is involved in RL. We designed an algorithm that learns faster by adaptively updating the reward value in the form of a real number derived from human evaluation. Then, the adaptively updated reward is used for learning by calculating the final reward, with the reward being updated as the learning object in the system environment.

Further, a basketball game was designed as an example in which multiple network topologies coexist in a complex form. A basketball game is a complex problem in which the network changes in real time and the objective is correct passing of the ball among players on the same team to score points.

The remainder of this paper is organized as follows. Section 2 reviews the existing research and literature on the application of RL in various fields, focusing on RL studies for effective control of robots and machines that simulate humans. Section 3 describes in detail the adaptive update strategy framework of the human evaluation reward with the SGD algorithm proposed in this paper. Section 4 discusses the implementation and experimental examples of the proposed algorithm and framework and compares this approach with the methods used in previous studies. Finally, Section 5 summarizes the conclusions and directions for future research.

2. Background and Literature Review

RL generally involves an advanced model of the Markov decision process (MDP). In terms of sequential decision making, it is based on the interaction between the current state and the system. In this method, the reward is calculated, and then the action needed to achieve the learning objective is determined [1].

The general learning method of the RL algorithm is the Q-learning method, which calculates and updates the Q-function, the behavior value function of the learning object, at every time t. At this time, the algorithm is designed to calculate the Q-function by maximizing the reward value [4]:

$$Q_{t+1}(s_t, a_t) = (1-\alpha)Q_t(s_t, a_t) + \alpha(r_{t+1} + \gamma max Q_t(s_{t+1}, a)) \quad (1)$$

In (1), s_t is the state at time t; a_t is the learning object behavior at time t; r_{t+1} is the reward value at time $t+1$; and α is the learning rate. The closer to α, the greater the value of the situation at the current time t and the behavior of the learning object. The discount rate γ is used to adjust the reward percentage for future behavior [5].

RL is used in a wide range of fields to solve important issues. This technique is applied according to the situation and environment, and methods of solving the corresponding problems are designed. In robot control, RL is an interesting topic and the most commonly used AI method. Research on various robotics topics, such as the use of robots in intelligence, soft robotics, and robot automation through navigation and autonomous control, has been conducted. Table 1 summarizes recent studies focusing on the relationship between humans and robots in relation to the learning of objects that are considered human interventions. It also lists the applications of these studies, keywords, and design methods of RL [6–11].

Table 1. Application and learning design methods related to robotics.

Research Studies	Application	Learning Method
Shastha [6]	Meal assistant robot to supply liquid in a cup	Comparison of five reinforcement learning (RL) algorithms (policy iteration, value iteration, Q-learning, state-action-reward-state-action (SARSA), deep Q network (DQN))
Lin [7]	Biped walking and balance control	Discrete-action Q-learning
Sheng [8]	Table-lifting task performed jointly by a human and robot	Q-learning
Kormushev [9]	1. Pancake flipping task 2. Bipedal walking energy minimization task 3. Archery-based aiming task	1. Dynamic movement primitives + RL 2. Evolving policy + RL 3. Regression
Wang [10]	Robot to learn from human demonstrations about assembly tasks	Maximum entropy inverse RL (human teaching robot using natural language)
Zhu [11]	1. Peg-in-hole task 2. Slide in the groove assembly task 3. Bolt-screwing task	Learning from demonstration

For example, RL has been effectively applied to enable individuals suffering from limb paralysis to drink liquids directly with the help of robotic manipulator arms [6]. In that study, five algorithms were applied and compared. Learning was performed effectively with a software emulator program, and the developed solution gave the user the ability to manipulate the cup. An assistant robot was effectively designed with the focus of supplying liquid from the cup using feedback through sensors that provided direct interaction between the human and robot.

Subsequently, RL was applied for stable, dynamic walking of biped robots [7]. This study was performed in the absence of prior knowledge or information on dynamic models, and the robot operation was controlled by mapping the motion space from the discrete to continuous domain. The research objective was to solve complex control problems. Among the components constituting the robot legs, a zero-moment point was selected from the sole and mapped to the movement of the limbs to learn balance. This study proved that a robot can learn how to improve its motion in terms of walking speed. Further, the proposed algorithm was implemented in a physical robot to prove its validity and effectiveness.

Another article suggested a framework that includes a learning phase that mimics human behavior and an RL phase that learns robot behavior [8]. The two-stage learning framework, which combines imitation and RL, shows how to work with people to lift tables quickly and successfully. The first stage is for learning the existence and location of the object called a table, and the second stage is the learning stage for performing operations and tasks. The robot operation is controlled by combining two types of controllers. This research demonstrated the successful construction of a collaborative robot designed to predict human movements and take proactive actions.

In another study, RL was applied to pancake flipping, energy minimization of bipedal walking robots, and archery-based aiming robots [9]. The authors argued that the ultimate goal of RL is to provide robots with the abilities to learn, improve, adapt, and play in tasks with dynamically changing constraints based on navigation and self-learning. It was suggested that RL is appropriate for highly dynamic tasks with clear scales and argued that imitation learning should be easy to demonstrate, use clear practices, and be effective for slow work. The regression-based learning algorithm was effective when the goal was small.

Further, the design of collaborative robots for assistance in assembly operations in manufacturing was investigated [10]. Collaborative robots are used in intelligent manufacturing-related environments

and are developed to learn from human demonstrations and support human partners in collaborative environments. According to the personal preferences of humans, natural language instructions can be used to teach robots. The robot learns from human demonstrations using the maximum entropy inverse RL algorithm, and the task-based learning is updated using the optimal assembly strategy. These studies have shown that RL can be effectively used in the design of human–robot collaboration.

Regarding detailed investigation of how to imitate human behavior, studies have been conducted on methods of demonstrating an example motion for a robot in assembly work and extracting a manipulation function for robot learning and motion imitation [11]. In one method, the robot can directly learn how to control its movement. In a second method, when designing a robotic arm, a motion sensor can be attached to a human arm to enable human behavior to be mimicked. Finally, remote operation and control boxes can be utilized to provide hints to a robot. Each method was used to establish a strategic method by direct or indirect human intervention for robot learning.

In addition to robot control, RL is applied and used effectively in various fields. For convenience in daily life, RL has been applied to drone delivery, home energy system optimization, autonomous driving, and automatic parking systems [12–15]. In Internet of Things devices and networks, RL is mainly used to control traffic and congestion in complex situations. To reduce the collisions between the system and client effectively, the access method is designed using rule-based algorithms and RL. RL is also utilized as a means of selecting the appropriate channel [16–19]. RL is applied to the problem of choosing a route to escape to a destination by avoiding obstacles.

Several existing research studies handling human-centered RL technologies have been applied to various applications. Kim and Lee [20] and Lee [21] applied RL techniques to several evacuation frameworks. In dynamic situations such as sudden obstacles or removals of exits, these frameworks generated evacuation routes considering humans' interactions and their congestions. Another application handling human-centered technologies and human–artificial hybrid intelligence is the bio-signal processing between human and a system with artificial intelligence modules. Kim et al. [22] analyzed both human–system interactions using a stimulus-producing electroencephalogram (EEG). In the research study, EEG signals are obtained in real-time and are used for evaluating human's satisfactions with the interface which a system with artificial intelligence modules provides. Moreover, this area of research is directly related to drone control problems, where RL has been applied to design drones with obstacle avoidance. The data obtained from the sensor module mounted on the drone are used to configure the environment and state of the RL model, and the drone is controlled by designing an algorithm to maximize the reward value obtained from operation [12,23]. RL is also used to design energy management systems to determine the balance between agents and optimal scheduling strategies. The RL algorithm is designed to achieve an optimal equilibrium of agent rewards for balanced energy distribution and scheduling [13,24]. Studies in which RL has been applied to large-scale social infrastructures such as ships and aircraft have mainly dealt with ship route planning, aircraft radar design, and aircraft detection systems. The route planning problem is often addressed in RL, utilizing an RL algorithm that yields the maximum reward value for an unmanned ship. In aircraft detection systems and radar designs, RL is applied to optimal radar system design and aircraft image analysis to detect radio waves and minimize unnecessary interference [25–27].

3. Adaptive Human Evaluation Strategy Framework Using the SGD-Based Reinforcement Learning

The present study is related to the basketball game problem, which involves competition between the two teams as shown in Figure 1. The reason for focusing on a basketball game in this study was to represent a network topology in which two independent states coexist in the same environment. In a competition between two teams, such as a basketball game, the interference of one team with the goal of the other team occurs because the two network topologies coexist, which is very appropriate for expressing the competition. The proposed framework considers a dynamic competing network where both human groups are competing with each other. As one of these characteristics is a volatile

environment, human's evaluations as well as machine learning techniques are essential. For this manner, the framework is proposed and tested seriously. In order to show the effectiveness of the proposed framework, a basketball game is illustrated.

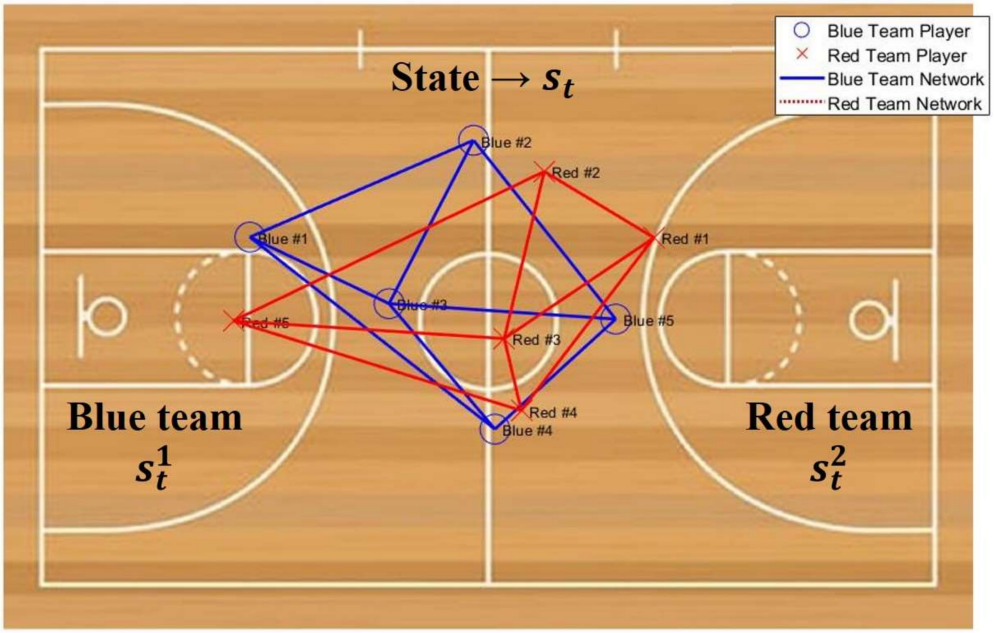


Figure 1. Basketball game and competitive network topologies.

A basketball game is very fast, and the learning environment is highly complex; therefore, appropriate RL techniques should be applied. In this study, the learning goal of the basketball game was to pass the ball to a member of the blue team, which was interrupted by the players on the red team. The network topologies of the players on the red and blue teams coexisted in the same environment. Unlike traditional RL problems, it deals with multiple complex network topologies, rather than a single topology.

Existing RL network problems consist of learning objectives involving a single network topology. However, a different method was necessary in this study, as it includes multiple network topologies that form a competing network topology in a dynamically changing state. When RL is applied in a single network topology, the reward policy can be learned through a system reward update computed from the learning object. This method is very simple, and the rewards that occur in a single network can be calculated and updated through actions in a given environment and current state.

In this report, we propose a method of establishing reward policies for complex networks that learn two network topologies in the same environment. This method involves updating the reward policy by applying the rewards obtained through human evaluation as well as the system rewards calculated from the learning target.

First, to address the RL problem, which consists of two complex network topologies, the Q-function is defined as (2), and the maximum human evaluation reward value is calculated, taking into account all time periods t:

$$Q_{t+1}\left(s_t^1,\ s_t^2, a_t\right) = (1-\alpha)Q_t\left(s_t^1,\ s_t^2, a_t\right) + \alpha\left(h_{t+1}^* + \gamma max Q_t\left(s_{t+1}^1,\ s_{t+1}^2, a_t\right)\right). \quad (2)$$

States s_t^1 and s_t^2 coexist and are affected by the same operation a_t in the same environment. In terms of network topology, states s_t^1 and s_t^2 are determined independently of each other in the coordinate system in which the node is located at the location.

However, the interference between the networks is affected by operation a_t. Therefore, it can be defined as (3):

$$a_{t+1}(\overrightarrow{s_t^2}) = s_{t+1}^2. \tag{3}$$

The important point here is that, unlike when the RL algorithm is applied to a single network topology, as in the existing research, the reward acquisition process is performed through human evaluation. At every time t, the state changes so that the learning object can learn with effective rewards, human intervention occurs, and the reward policy is evaluated accordingly. In previous studies [2,3], the learning object has been taught using a binary human evaluation method to solve complex problems through human intervention and evaluation.

However, in this study, we designed a human evaluation algorithm by emphasizing that human evaluation should not be simply performed using a binary process to solve complex network topology and that human evaluation should involve quantitative, real number feedback.

The human evaluation reward her_t obtained during learning at all times t can be modeled as a Gaussian distribution, as shown in (4):

$$her_t \sim N\left(\mu, \sigma^2\right), \tag{4}$$

where μ is the mean value of the evaluation, and σ is the standard deviation. In general, the mean of the Gaussian distribution can be estimated as $\overline{her_t}$, using (5):

$$\overline{her_t} = \frac{\sum_{i=1}^{t} her_i}{n}, \tag{5}$$

where n is the sample size and the number of quantitative rewards from human evaluation learned over all times t. The standard deviation of the Gaussian distribution can be estimated using (6):

$$\hat{\sigma} = \sqrt{\frac{n\sum_{i=1}^{t} her_i{}^2 - \left(\sum_{i=1}^{t} her_i\right)^2}{t(t-1)}}. \tag{6}$$

In this study, human evaluation was performed by repeated learning, and the SGD algorithm was used to update $\overline{her_t}$ adaptively. The complex network topologies covered in this study are computationally expensive due to the large number of human evaluations in the learning process. The general form of the SGD algorithm used in this study is shown in (7):

$$\overline{her_{t+1}} = \overline{her_t} - \eta \Delta F\left(\overline{her_t}\right). \tag{7}$$

The human evaluation value estimated through repeated learning is called $\overline{her_t}$, and the difference between $\overline{her_t}$ updated at the present time t and $\overline{her_{t+1}}$ at the next environmental time point $t+1$ is defined as the loss function $F\left(\overline{her_t}\right)$. A slope is used to minimize this function. Iteration is performed over the time t by a certain amount in the opposite direction of the gradient to find the value of $\overline{h_t}$ that minimizes $F\left(\overline{her_t}\right)$. This change equation is defined by (7).

η is a predetermined step size. In general, the use of all of the data to calculate $F\left(\overline{her_t}\right)$ is called batch gradient descent. However, this calculation requires excessive computation because $F\left(\overline{her_t}\right)$ must be calculated for all of the data in one step. In this study, the computational complexity was higher than that in general problems because the two network topologies involved complex and special problems.

To prevent this problem, a method called SGD was used. In this method, $F\left(\overline{her_t}\right)$ is calculated only for some small collections of data instead of all of the data. Because this method is much faster, more steps can be performed in the same time, and if the process is repeated several times, it usually converges to the same result as the batch. It is also possible to use SGD to converge in a better direction

without falling into the local minima that will be lost in the batch gradient descent. In this study, SGD was used to update the estimated human value in repetitive learning, as shown in Figure 2. To evaluate the two network topologies that compete in a complex manner in repetitive learning, humans score points in the form of real numbers.

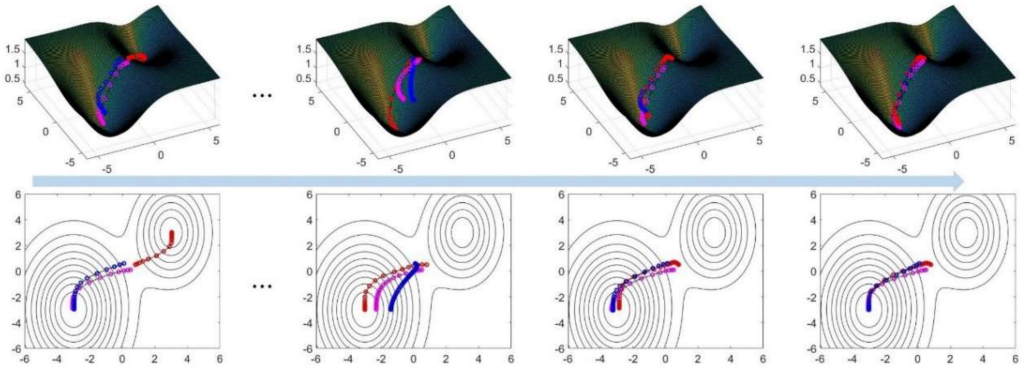


Figure 2. Human evaluation reward update using a stochastic gradient descent (SGD) algorithm in the repeated learning process.

In the iterative learning, the values evaluated by humans in real form were updated as shown in Figure 2 using the SGD algorithm. In the early stages, however, these values converge to the local minima. To solve this problem, an adaptive SGD algorithm was used. The step size N was set differently for each estimation iteration, as shown in (8):

$$E_{t+1} = E_t + \left(\eta \Delta F\left(\overline{her_{t+1}}\right)\right)^2 \tag{8}$$

$$\overline{her_{t+1}} = \overline{her_t} - \frac{\eta}{\sqrt{E_t + \epsilon}} \Delta\eta \Delta F\left(\overline{her_t}\right). \tag{9}$$

Therefore, if the variation of the estimated human evaluation reward value is small, η increases, and if it is large, η decreases. E_{t+1} is a function that updates the sum of squares of the gradient through which $\overline{her_t}$ moves in time t. When $\overline{her_t}$ is updated as in (9), $\overline{her_t}$ moves in inverse proportion to the root value of $E_t + \epsilon$ in the existing step size η. It means that if a step size η becomes larger, $\overline{her_t}$ moves considerably. Since this adaptive method moves by setting the step size differently for each $\overline{her_t}$, it is highly likely to approach the optimum when the state of the environment evaluated by humans appears frequently or under the same conditions; hence, the fine value is adjusted while moving to small step sizes. Lesser variation of $\overline{her_t}$ is designed to increase the step size to reach the optimum value. This method involves moving in a direction such that the loss can be reduced quickly and is a strategic and effective method of updating the human evaluation in the competitive problem of network topology coexisting in complex environment.

As shown in (10), her_t is updated again by adopting the maximum of $\overline{her_t}$ and her_t:

$$her'_t = \max\left(\overline{her_t}, her_t\right). \tag{10}$$

The correction value her'_t adaptively updated by human evaluation must be calculated appropriately with the reward value r_t of the system reward derived by updating the learning object to determine the final reward value h^*_t. Note that h^*_t is the reward of learning finally used for repetitive learning, and her'_t is the reward calculated by human intervention and evaluation during the learning process.

$$h^*_t = (1 - \delta) r_t + \delta her'_t \tag{11}$$

To design human interventions and evaluations adaptively in an iterative RL process, SGD algorithms are used to implement reward policies and to perform appropriate calculations with rewards

computed within competing systems with complex coexisting network topologies. A framework that summarizes these interactions is shown in Figure 3.

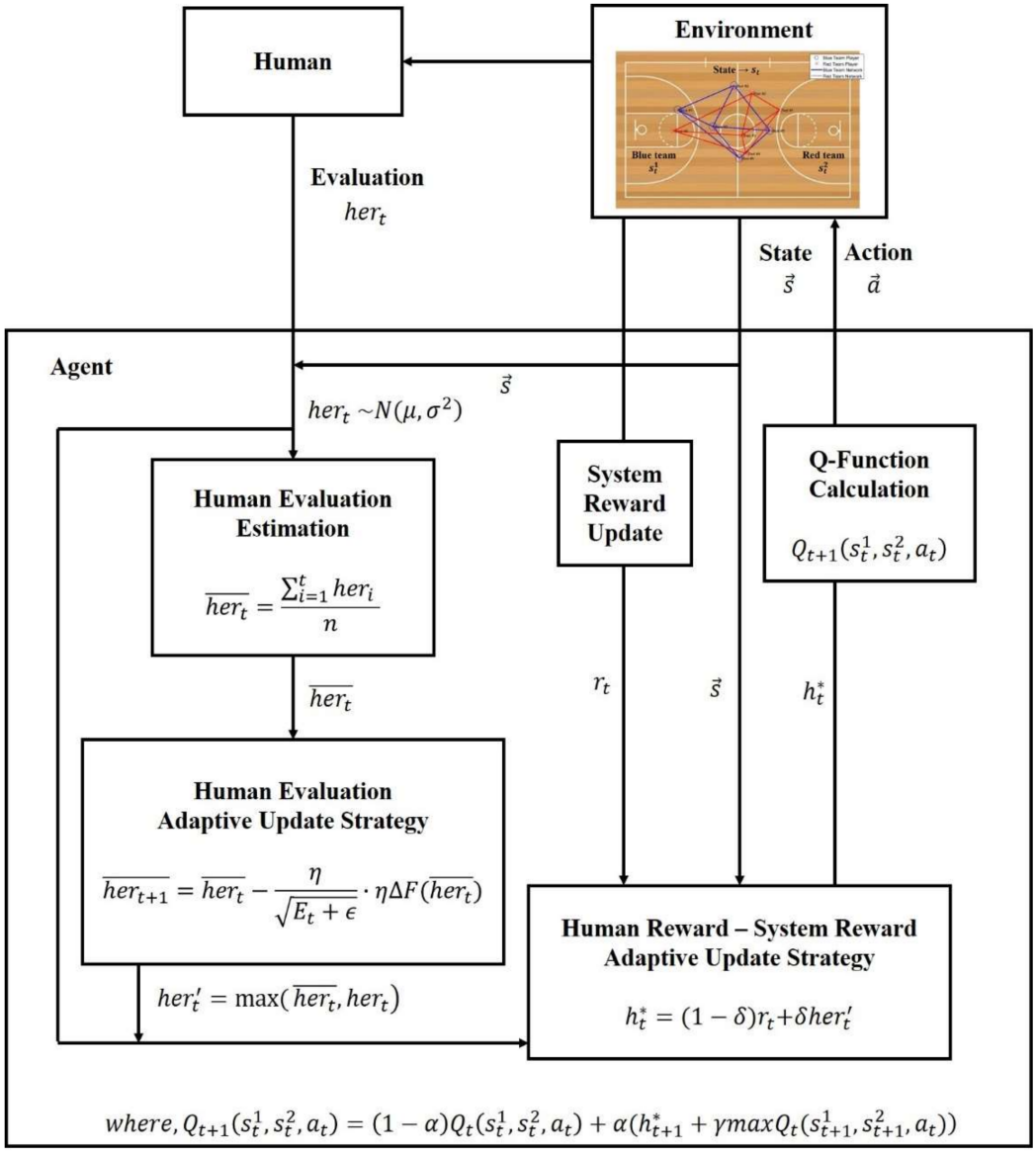


Figure 3. Adaptive human evaluation strategy reinforcement learning (RL) framework using an SGD algorithm.

In this paper, we propose a method of effectively updating rewards in a complex network topology. To set and update the rewards in the RL process, quantitative human evaluation is performed in real form and the reward policy is updated using the adaptive SDG algorithm. Afterwards, the system implements the rewards and appropriate calculations, and the RL model is designed in a more advanced way. Algorithm 1 details the overall algorithm of this framework.

Algorithm 1. RL algorithm using adaptive human evaluation reward updating to establish reward policies.

$\delta \leftarrow$ constant: human intervene rate
$\gamma \leftarrow$ constant: discount rate
$\alpha \leftarrow$ constant: learning rate
for $1 \leq t \leq T$
  while true **do**
    $\vec{s} \leftarrow getState()$
    $TakeAction(\vec{a})$
    $TakeSystemReward(r_t)$
    wait for next time step
    $her \leftarrow gethumanEvaluationFeedback()$
    if $her = !0$
      Estimation for $her \sim N(\mu, \sigma^2)$
      $\overline{her_t} = \frac{\sum_{i=1}^{n} her_i}{n}$
      $her_{t+1} = her_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \Delta F(her_t)$
      $her'_t = \max(\overline{her_t}, her_t)$
        $TakeFinalReward(h^*_t)$
        $h^*_t = (1-\delta) r_t + \delta her'_t$
          $Q_{t+1}(s^1_t, s^2_t, a_t) = (1-\alpha) Q_t(s^1_t, s^2_t, a_t) + \cdots$
        $\alpha(h^*_{t+1} + \gamma max Q_t(s^1_{t+1}, s^2_{t+1}, a_t))$
    end if
  end while
end for

4. System Implementation and Experimental Results

This section describes in detail the implementation of the proposed adaptive human evaluation strategy framework and presents the numerical analysis performed using software programs. The software program implemented as shown in Figure 4 consists of six different functional panels. Table 2 summarizes the functions of each panel.

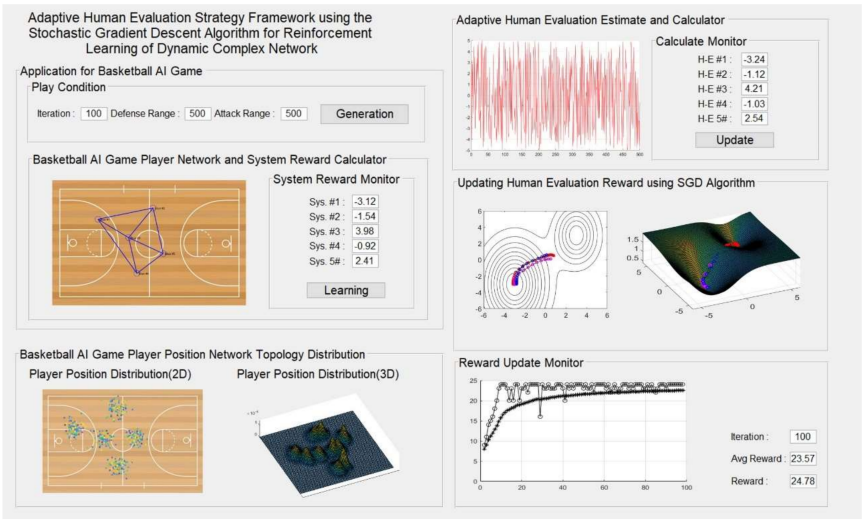


Figure 4. Adaptive human evaluation strategy framework software program using the SGD algorithm for RL of a dynamic complex network.

Table 2. Function summary of adaptive human evaluation strategy framework implementation using the SGD algorithm for RL of dynamic complex network.

Panel Name	Detailed Function	Configurations
Application for basketball AI game	Define an iteration and defense, attack range	- Iteration - Defense range - Attack range
Basketball AI game player network and system reward calculation	Check the network of attacking players and calculate and show system rewards	- Network topology of attacking players - System reward monitor
Adaptive human evaluation estimation and calculation	Distribution of position changes according to the repetition learning time point.	- Player position distribution (2D) - Player position distribution (3D)
Adaptive human evaluation estimation and calculation	Enter the reward by evaluating the condition through human intervention in repetitive learning. The reward entered is estimated and mapped to a normal distribution.	- Human evaluation reward estimate chart - Adaptive human evaluation calculation monitor
Updating human evaluation reward using SGD algorithm	Update human evaluation reward adaptively using SGD algorithm	- Chart of updating human evaluation reward using SGD algorithm
Reward update monitoring	Derivation of the reward value obtained as a result of learning for each iteration.	- Chart of analyzing integrated reward - Iteration, average reward - Reward at time t

The first panel is called "Application for basketball AI game," where the artificially coexisting and dynamically changing network topology problem discussed in this paper is applied to an AI basketball game. To determine the basketball game conditions, the number of times to repeat the lesson and the ranges of the attacking and defending teams are determined.

The second panel, called "Basketball AI game player network and system reward calculation," shows the basketball team network fluctuating dynamically during repeated learning, while calculating the system reward using the general RL algorithm.

The third panel, "Basketball AI game player position network topology distribution," depicts the position distribution of basketball players on the two-dimensional (2D) plane in the repetitive learning, as well as the three-dimensional (3D) mesh. Each change is shown in detail in Figure 5, and the cumulative changes as the learning is repeated are evident.

The fourth panel is called "Adaptive human evaluation estimation and calculation," where the evaluation is made through human intervention when the players of the attacking team choose the direction in which to pass the ball, and the reward is given as a real number between –5 and 5. The user directly enters the number in the form of a real number into the software system. The human evaluation rewards evaluated in this manner are handled more effectively in the fifth panel.

In the fifth panel, "Updating human evaluation reward using SGD algorithm (in repeated learning process)," the SGD algorithm is used adaptively to update the reward value that the human user entered through evaluation.

Finally, as shown in Figure 6, the system and adaptive human evaluation rewards calculated in the second and fifth panels are appropriately calculated to derive the final reward value and proceed with the learning. This final sixth panel is called "Reward update monitoring" and shows the rewards and average rewards in repeated learning of basketball games depicted using a complex and dynamically changing network with the proposed framework.

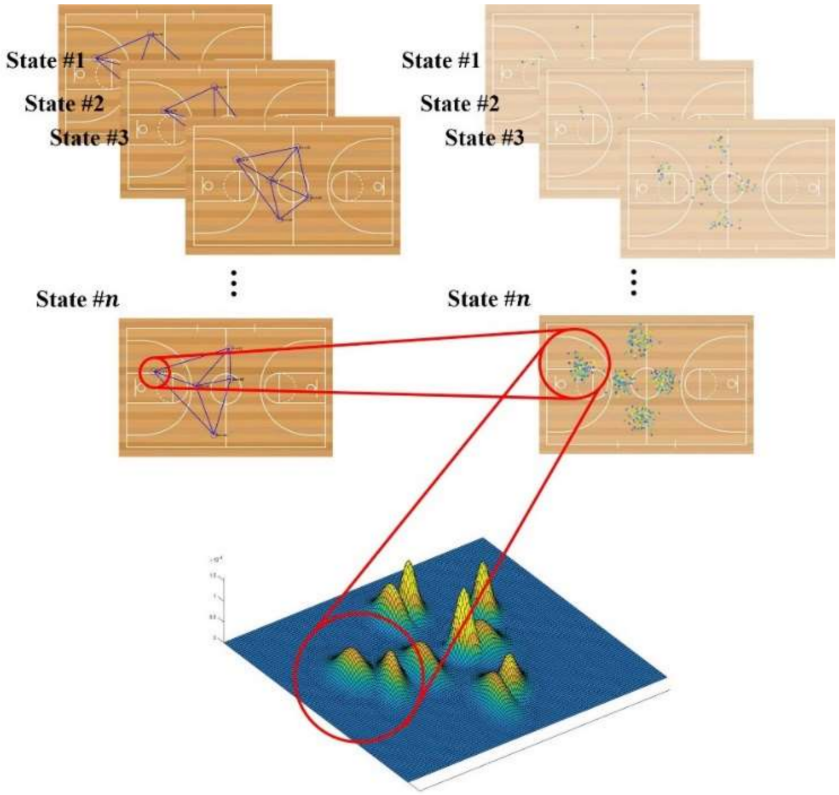


Figure 5. Player position network topology (2D and 3D) in basketball game problem.

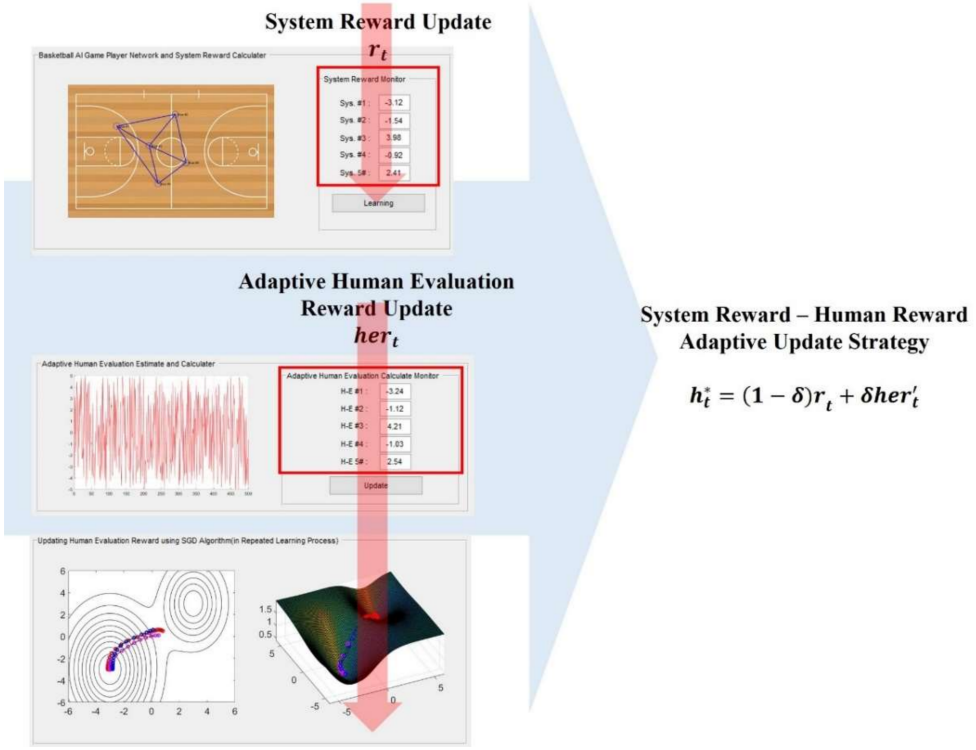


Figure 6. Process of calculating the final reward strategically using the SDG algorithm.

Figure 7 compares the RL methods using the adaptive human evaluation strategy proposed in this paper with those proposed in previous studies. The existing methods that were compared with the

proposed method were human evaluation with binary updating, the SARSA method, and the MDP method. Table 3 defines and shows experimental conditions that apply equally to all methods. All of the methods involve learning basketball games with the complex and dynamically changing network topologies discussed in Section 3.

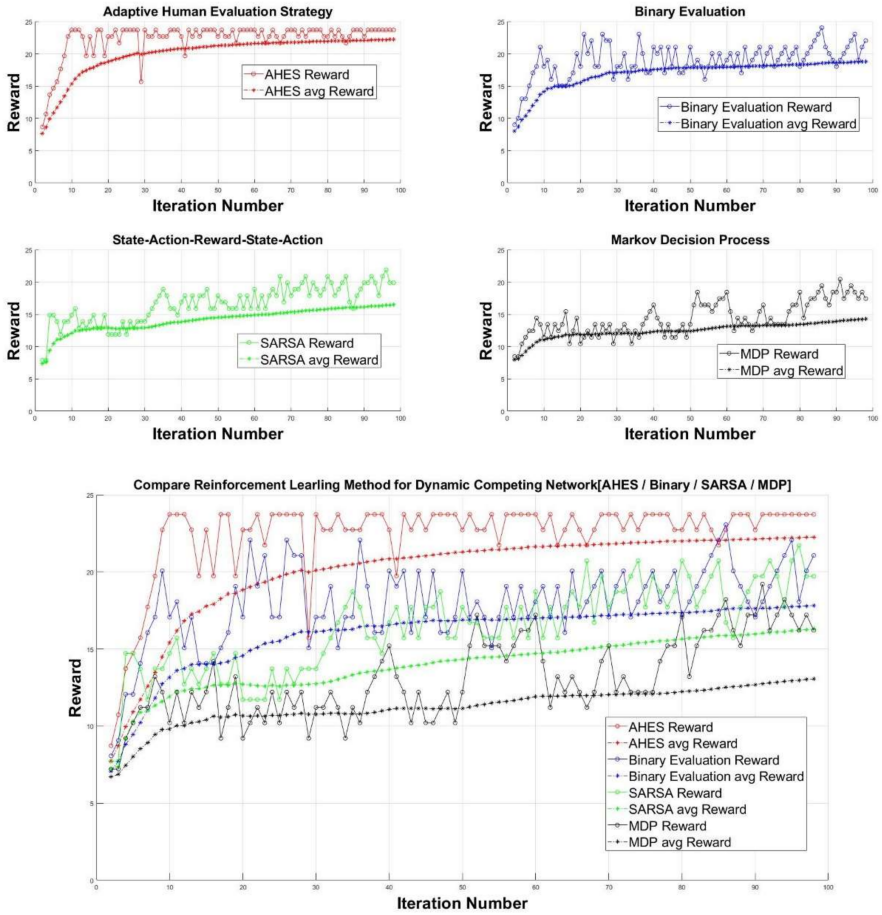
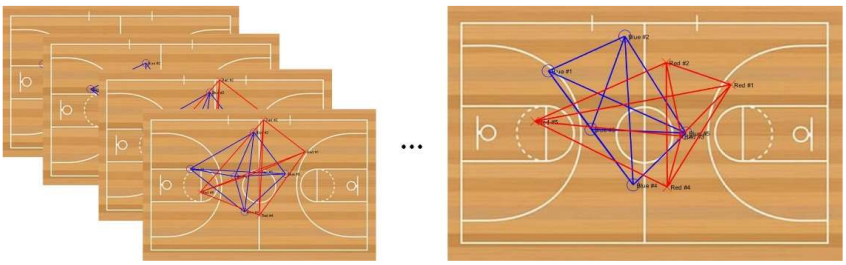


Figure 7. Comparison of four RL methods for complex and dynamically changing network topologies.

Table 3. Definition of experimental conditions for complex and dynamically changing network topologies.

Experimental Conditions	
	
Network condition	Number of networks: 2 Number of nodes: 5 in each network
Iteration	100
Discount rate	0.9
Learning rate	0.9

As shown in Table 4, the type and human intervention rate of human evaluation of each method, the average reward value, the Q-function value was calculated, and the points of convergence with the maximum reward value were compared. As shown in Table 4, the RL method with the proposed adaptive human evaluation strategy achieved convergence with the highest maximum reward value. In addition, it exhibits the fastest convergence to the maximum reward in Figure 7. This result demonstrates that, unlike when the method of updating the system reward using the existing learning objects is used, the learning reaches the maximum reward value faster when the quantitative evaluation is performed through human intervention.

Table 4. Comparison of experimental results of four RL methods.

	Adaptive Human Evaluation Strategy	**Binary Evaluation**	**SARSA**	**MDP**
Human evaluation	Real number	Binary number	No intervention	No intervention
Human intervention rate	$\delta = 0.8$	$\delta = 0.8$	No intervention	No intervention
Average reward	23.57	17.72	16.01	13.54
Q-function value	24.78	21.51	19.14	16.12
Convergence time of maximum reward value	$t = 9$	$t = 87$	$t = 97$	$t = 92$

5. Conclusions

RL has been studied in various forms to train learning objects to achieve desired goals. It is proposed to design an algorithm to apply RL by mapping an environment with high complexity and time-sensitive dynamic changes to the network topology.

To learn dynamically changing network topologies effectively, we designed a system to evaluate the status and update the rewards through human intervention. Unlike in the existing methods, quantitative and clear evaluations are made using real numbers. In addition, the reward value evaluated from human intervention is calculated by applying the SDG algorithm to establish an adaptive update strategy to improve the learning speed. This RL method is stable and effective and enables accurate reward updating through human intervention. After that, the system rewards and adaptive rewards estimated from the human evaluations are calculated and updated accordingly. To demonstrate the effectiveness of this technique, a basketball game was mapped to a competing network topology and investigated experimentally. The proposed framework was compared with the existing RL methods (binary evaluation, the SARSA method, and the MDP method) in the software environment. The proposed adaptive human evaluation strategy converged to the maximum reward value the fastest and produced a high Q-function value.

In future research, methods of effectively learning two or more opposing objects in a physical environment should be considered, as interventions that deliver human evaluations directly to learning objects in physical environments must be designed with more sophisticated and advanced reward updating strategies.

Author Contributions: Methodology, J.K. and H.L.; software, J.K.; validation, J.K. and H.L.; formal analysis, J.K.; investigation, J.K.; writing—original draft preparation, J.K.; writing—review and editing, J.K. and H.L.; visualization, J.K.; supervision, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003).

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

References

1. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
2. Knox, W.B.; Stone, P. Tamer: Training an agent manually via evaluative reinforcement. In Proceedings of the 2008 7th IEEE International Conference on Development and Learning, Monterey, CA, USA, 9–12 August 2008; pp. 292–297.
3. Celemin, C.; Ruiz-del-Solar, J. COACH: Learning continuous actions from corrective advice communicated by humans. In Proceedings of the 2015 International Conference on Advanced Robotics, Turkey, Istanbul, 27–31 July 2015; pp. 581–586.
4. Greenwald, A.; Hall, K.; Serrano, R. Correlated Q-learning. In Proceedings of the International Conference on Machine Learning, Wachington, DC, USA, 21–24 August 2003; p. 242.
5. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
6. Kumar Shastha, T.; Kyrarini, M.; Gräser, A. Application of Reinforcement Learning to a Robotic Drinking Assistant. *Robotics* **2020**, *9*, 1. [CrossRef]
7. Lin, J.L.; Hwang, K.S.; Jiang, W.C.; Chen, Y.J. Gait balance and acceleration of a biped robot based on Q-learning. *IEEE Access* **2016**, *4*, 2439–2449. [CrossRef]
8. Sheng, W.; Thobbi, A.; Gu, Y. An integrated framework for human–robot collaborative manipulation. *IEEE Trans. Cybern.* **2014**, *45*, 2030–2041. [CrossRef]
9. Kormushev, P.; Calinon, S.; Caldwell, D.G. Reinforcement Learning in Robotics: Applications and Real-World Challenges. *Robotics* **2013**, *2*, 122–148. [CrossRef]
10. Wang, W.; Li, R.; Chen, Y.; Diekel, Z.M.; Jia, Y. Facilitating Human–Robot Collaborative Tasks by Teaching-Learning-Collaboration from Human Demonstrations. *IEEE Trans. Autom. Sci. Eng.* **2018**, *16*, 640–653. [CrossRef]
11. Zhu, Z.; Hu, H. Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robotics* **2018**, *7*, 17.
12. Muñoz, G.; Barrado, C.; Çetin, E.; Salami, E. Deep Reinforcement Learning for Drone Delivery. *Drones* **2019**, *3*, 72. [CrossRef]
13. Lee, S.; Choi, D.-H. Reinforcement Learning-Based Energy Management of Smart Home with Rooftop Solar Photovoltaic System, Energy Storage System, and Home Appliances. *Sensors* **2019**, *19*, 3937. [CrossRef]
14. Hu, B.; Li, J.; Yang, J.; Bai, H.; Li, S.; Sun, Y.; Yang, X. Reinforcement Learning Approach to Design Practical Adaptive Control for a Small-Scale Intelligent Vehicle. *Symmetry* **2019**, *11*, 1139. [CrossRef]
15. Zhang, P.; Xiong, L.; Yu, Z.; Fang, P.; Yan, S.; Yao, J.; Zhou, Y. Reinforcement Learning-Based End-to-End Parking for Automatic Parking System. *Sensors* **2019**, *19*, 3996. [CrossRef]
16. Ma, J.; Hasegawa, S.; Kim, S.-J.; Hasegawa, M. A Reinforcement-Learning-Based Distributed Resource Selection Algorithm for Massive IoT. *Appl. Sci.* **2019**, *9*, 3730. [CrossRef]
17. Lee, T.; Jo, O.; Shin, K. CoRL: Collaborative Reinforcement Learning-Based MAC Protocol for IoT Networks. *Electronics* **2020**, *9*, 143. [CrossRef]
18. Chen, J.; Chen, S.; Wang, Q.; Cao, B.; Feng, G.; Hu, J. iRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks. *IEEE Internet Things J.* **2019**, *6*, 7011–7024. [CrossRef]
19. Qiu, X.; Liu, L.; Chen, W.; Hong, Z.; Zheng, Z. Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8050–8062. [CrossRef]
20. Kim, J.; Lee, H. Multi-agent Reinforcement Learning based Evacuation Framework Considering Both Evacuation Time and Crowdedness. *J. Korean Inst. Intell. Syst.* **2016**, *26*, 335–342.
21. Lee, H. Human crowd evacuation framework and analysis using look-ahead-based reinforcement learning algorithm. *Int. J. Digit. Hum.* **2016**, *1*, 248–262. [CrossRef]
22. Kim, J.; Kim, S.; Lee, H. Pattern Recognition and Classifier Design of Bio-signals based Interface in Human-Artificial Intelligence Interaction (HAII) Framework for Real Time Evaluation of Emotions. *J. Korean Inst. Intell. Syst.* **2019**, *29*, 335–342.
23. Shin, S.-Y.; Kang, Y.-W.; Kim, Y.-G. Obstacle Avoidance Drone by Deep Reinforcement Learning and Its Racing with Human Pilot. *Appl. Sci.* **2019**, *9*, 5571. [CrossRef]
24. Fang, X.; Wang, J.; Song, G.; Han, Y.; Zhao, Q.; Cao, Z. Multi-Agent Reinforcement Learning Approach for Residential Microgrid Energy Scheduling. *Energies* **2020**, *13*, 123. [CrossRef]
25. Guo, S.; Zhang, X.; Zheng, Y.; Du, Y. An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning. *Sensors* **2020**, *20*, 426. [CrossRef] [PubMed]

26. Wu, Q.; Wang, H.; Li, X.; Zhang, B.; Peng, J. Reinforcement Learning-Based Anti-Jamming in Networked UAV Radar Systems. *Appl. Sci.* **2019**, *9*, 5173. [CrossRef]
27. Li, Y.; Fu, K.; Sun, H.; Sun, X. An Aircraft Detection Framework Based on Reinforcement Learning and Convolutional Neural Networks in Remote Sensing Images. *Remote Sens.* **2018**, *10*, 243. [CrossRef]