Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey

Mohamed Said Frikha, Sonia Mettali Gammar, Abdelkader Lahmadi, Laurent Andrey

Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey

Mohamed Said Frikha[a], Sonia Mettali Gammar[a], Abdelkader Lahmadi[b], Laurent Andrey[b]

[a]*CRISTAL Lab, National School of Computer Science, University of Manouba, Manouba, Tunisia*
[b]*CNRS, Inria, LORIA, Université de Lorraine, F-54000 Nancy, France*

Abstract

Nowadays, many research studies and industrial investigations have allowed the integration of the Internet of Things (IoT) in current and future networking applications by deploying a diversity of wireless-enabled devices ranging from smartphones, wearables, to sensors, drones, and connected vehicles. The growing number of IoT devices, the increasing complexity of IoT systems, and the large volume of generated data have made the monitoring and management of these networks extremely difficult. Numerous research papers have applied Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) techniques to overcome these difficulties by building IoT systems with effective and dynamic decision-making mechanisms, dealing with incomplete information related to their environments. The paper first reviews pre-existing surveys covering the application of RL and DRL techniques in IoT communication technologies and networking. The paper then analyzes the research papers that apply these techniques in wireless IoT to resolve issues related to routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and caching. Finally, a discussion of the proposed approaches and their limits is followed by the identification of open issues to establish grounds for future research directions proposal.

Keywords: Internet of Things, Reinforcement learning, Deep reinforcement learning, Wireless Networks.

1. Introduction

Internet of Things has introduced more openness and complexity by connecting a large number and a variety of wireless-enabled devices. It allows the collection of a massive amount of data and appliance of control actions in various applications. [1, 2] such as healthcare [3], traffic congestion [4], agriculture [5], autonomous vehicle [6]. Wireless IoT devices such as connected vehicles, wearables, drones, and sensors, are able to interact with each other over the Internet, making people's life more comfortable. The deployment of these devices forms a set of Wireless Sensor Network (WSN), characterized by a large number of low-cost and low-power sensors with a short-range wireless transmission. WSN represents the primary source for collecting and monitoring information subsequently processed by the IoT due to its low cost and ease of integration.

Wireless technology has changed the way Internet Protocol (IP) devices communicate and share information using transmission through radio frequencies, light-waves, etc. These technologies also make it possible to deploy IoT networks in unreachable areas where it is unmanageable to build wired networks. Furthermore, various wireless technologies applied in IoT have been developed in the past few years to meet the requirements, including reducing energy consumption, compressing data overhead, and improving security and transmission efficiency for different networks. However, managing heterogeneous infrastructure is a complicated task, especially when dealing with large-scale IoT systems, and requires a complex system to ensure their operations and optimize data flow distribution.

Over the last five years, Machine Learning (ML) [7] techniques are adopted to integrate more autonomous decision-making in wireless IoT networks to effectively address their various issues and challenges, such as energy efficiency, load balancing, and cache management. Machine learning algorithms are also applied for IoT data analytics to discover new information, predict future insights, and offer new real-time services. Compared to statistical methods, ML algorithms provide better accurate predictions when dealing with very large data sets and they do not require heavy assumptions such as linearity or the distribution of variables [8]. They also have acceptable performance when using them for online classification or prediction [9].

ML techniques, especially Reinforcement Learning [10], attempt to make IoT nodes take self-decisions for multiple networking operations, including routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and caching. The RL agent must be able to understand the environment dynamics, without any prior knowledge, through the collected data and take the best action to achieve a networking goal, like reducing energy consumption, improving security level, and changing transmission channel.

However, as the complexity of the IoT networks increases with high-dimensional states and actions, traditional RL techniques show their limits regarding computation complexity and convergence towards a poor policy. Thus, Deep Reinforcement Learning techniques, a combination of RL and Deep Learning (DL) approaches [11, 12] based on Artificial Neural Networks (ANN) [13], are developed to overcome such limitations and make the learning and the decision operations more efficient.

In recent years, several surveys related to the integration of machine learning techniques in networks and IoT applications have been published. We provide in table 1 a summary of existing papers in this area. Papers [14, 15, 16] focused on the application of traditional RL in wireless networks. This technique can predict the near-optimal policy in cases where the number of states and actions is limited and improves the performance of networks with limited resources. The application of DRL has been discussed in [17, 18, 19] for IoT and real-world problems like cloud computing, autonomous robots, and smart vehicles. The studied papers in these surveys use DRL to accelerate the learning process, compared to the traditional RL, and reduce the storage space required by vast possible actions. Most surveys that study articles applying Machine Learning in networking focus on supervised and unsupervised methods [20, 21]. Only the survey proposed by [22] includes RL algorithms in its studied papers.

This paper provides a survey and a taxonomy of existing research which applies numerous RL and DRL algorithms for wireless IoT systems based on IoT application and network issues. An overview of these learning techniques is presented, and the main characteristics of the RL elements (i.e., state, action, and reward) are summarized Finally, we highlight the lessons learned with the remaining challenges and open issues for future research directions. To the best of our knowledge, there does not exist an article in the literature that is dedicated to surveying the application of reinforcement learning methods in different wireless IoT technologies and applications. Our survey differs from previous ones by covering the two RL and DRL methods for solving the network and application problems in wireless IoT, as proposed in research papers published in the period 2016-2020. We reviewed papers that address the following key issues: routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and edge caching.

The rest of this paper is organized as follows. Section 2 presents an overview of some wireless IoT networks. Section 3 introduces a brief description of the principle of RL, Markov Decision Process (MDP), and DRL. Section 4 reports research works that have applied RL and DRL techniques in the IoT environment according to their objective. A discussion is provided in Section 5 with statistics information on the articles reviewed in this work. Limitations of RL and DRL techniques and open challenges are identified in Section 6. Finally, Section 7 concludes this study.

2. Wireless IoT Systems

In this section, we provide an overview of wireless IoT systems studied in the surveyed papers and identifies their characteristics and challenges.

2.1. Wireless Sensor Network (WSN)

The evolution in the field of wireless communications and electronics has allowed building small and individual nodes, called sensors, that interact with the environment by sensing and controlling physical parameters, such as temperature, pressure, and motion. WSNs are composed of low-cost and battery-powered nodes that can send/receive messages to/from the sink and interact with each other through short-distance wireless communication. Using WSN networks has increased with the advent of the IoT since it is one of the big data sources for collecting and monitoring information further processed by the IoT. However, it has several weaknesses, including limited processing power, smaller memory capacities, and energy constraint with the difficulty of recharging or replacing the battery. In addition, some problems are related to the management of WSN that limit the technologies in which used, such as deployment, routing, security, and network lifetime.

2.2. Wireless Body Area Network (WBAN)

WBAN is a small, short-range, low-power network with a dozen sensors attached to or implanted in/around the human body. The sensor nodes are employed to detect physiological phenomena and provide real-time patient monitoring of different physical parameters, such as body temperature, blood pressure, and electrocardiography (heart rate). Using wireless communication, the collected information is then transmitted to a coordinator (i.e., sink node) that will process it, make decisions or raise an alert. Depending on the WBAN services for medical or non-medical applications, the communication characteristics vary. Different wireless technologies are applied in WBAN, including Bluetooth Low Energy (BLE), IEEE 802.15.4 (ZigBee), and IEEE 802.15.6 (Ultra Wide-Band) [23]. The WBAN standard specified for short communication range, low transmission power in particular for nodes under the skin, a minimal latency period especially for medical applications, and also supports mobility due to body movements [24].

2.3. Underwater Wireless Sensor Network (UWSN)

UWSN [25] is a self-configuring network that contains several autonomous components, such as vehicles and sensors, distributed underwater to perform environmental monitoring and ocean sampling tasks, like pressure, depth, visual imaging, assisted navigation, etc. The underwater architecture can be classified into two common categories: a two-dimension architecture, where a group of nodes is connected to one or more fixed anchor nodes, and a three-dimension architecture, where the sensors float at different depths. Due to water currents, the network nodes are dynamic, and the connectivity can vary with time. Due to water currents, network nodes are dynamic and connectivity can vary over time. Compared to the terrestrial environment, the performance of submarine sensors faces different challenges related to the physical nature that limit the bandwidth, leading to a high propagation delay, raise resource utilization, etc. For that, three principal wireless communication technologies for underwater environments are used, with different characteristics, depending on services requirements: optical, radio-frequency, and acoustic [26].

Table 1: Related surveys on the use of RL/DRL/ML in communication networks

Survey	Year	Contribution	Application domain	ML	RL	DRL
Al-Rawi *et al.* [14]	2015	The authors in this paper provided an overview of the application of RL based routing schemes in distributed wireless networks. The challenges, the advantages brought, and the performance enhancements achieved by the RL on various routing schemes have been identified.	Wireless networks - Routing		✓	
Cui *et al.* [20]	2018	The authors in this survey provided an overview of the ML techniques and solutions, in particular, the supervised and unsupervised solutions for IoT.	IoT	✓		
Althamary *et al.* [15]	2019	The paper summarized some issues related to vehicular networks and reviews the applications using the Multi Agent Reinforcement Learning (MARL) that enables decentralized and scalable decision making in shared environments.	Vehicular networks - MARL		✓	
Wang *et al.* [16]	2019	A classification of the Dynamic Spectrum Access (DSA) algorithms based on RL in cognitive radio networks has been presented.	Cognitive radio networks - DSA		✓	
Luong *et al.* [17]	2019	A comprehensive literature review on techniques, extensions, and applications of DRL to solve different issues in communications and networking has been surveyed in this paper.	Communications and networking			✓
Da Costa *et al.* [21]	2019	The security methods in terms of intrusion detection, for IoT and its corresponding solutions using ML techniques, have been studied in this work.	IoT - Security	✗		
Kumar *et al.* [22]	2019	The authors of this paper reviewed ML-based algorithms (includes RL methods) for WSNs, covering the period from 2014-March 2018. The different issues in WSNs and the advantages of selecting an ML technique in each case have been presented.	WSN	✓	✓	
Lei *et al.* [18]	2020	This paper initially described the general model of Autonomous Internet of Things (AIoT) systems based on the three-layer (i.e., perception layer, network layer, and application layer) structure of IoT. The classification of DRL AIoT applications and the integration of the RL elements for each layer have been summarized.	Autonomous IoT		✓	✓
Nguyen *et al.* [19]	2020	An overview of different technical challenges in multi-agent learning has been present, as well as their solutions and applications to solve real-world problems using DRL methods.	Multiagent Systems			✓

Legend: ✓ Covered; ✗ Partially Covered

2.4. Internet of Vehicle (IoV)

The integration of Vehicular Ad-Hoc Network (VANET) [27] into the IoT was an important milestone on the way to the advent of the IoV [28]. It refers to a network of different entities regarding vehicles, roads, smart cities, pedestrians that exchanged real-time information among them efficiently and securely. IoV has brought new applications to driving and using vehicles, for instance, safe and autonomous driving, crash response, traffic control, infotainment. Vehicular networks have various characteristics that differ from mobile node networks as a dynamic height topology since vehicles move at high speed and in a random way, a large-scale network with a variable density depending on the traffic situation, and without a constraint on computing power or energy consumption. Many special requirements therefore arise, includes processing big data using cloud computing, providing good connection links with an unstable mobile network, and achieving height reliability, security, and privacy of information. In terms of connectivity, the IoV consists of two types of wireless communications: Vehicle-to-Vehicle used to inter-vehicular exchanged information using, for example, the IEEE 802.11p [29] standard, and Vehicle-to-Infrastructure, also now as Vehicle-to-Road, where the vehicle exchanges information with roadside equipment with long-distance and high-scalability wireless technologies.

2.5. Industrial Internet of Things (IIoT)

The Industrial IoT [30] system refers to the adoption of the IoT framework for a large number of devices and machines in industrial sectors and applications. Machine-to-Machine (M2M) communication is a key technology in IIoT that allows devices to communicate and exchange data with each other. These intelligent machines can operate the highest level of automation without needing or with very minimal human intervention. The goal of IIoT is to achieve high operational efficiency, limit human errors, increase productivity, and reduce operating costs in both time and money, as well as predictive maintenance, using data collected from machines. The heterogeneity of the various communication protocols used and the complex nature of the system pose many challenges for designing wireless protocols for IIoT, such as higher levels of safety and security, efficient management of big data, and ensuring real-time communication in critical industrial systems.

3. Overview of Reinforcement Learning

This section provides a comprehensive background on different RL methods, including the principle of Markov Decision Process, Partially Observable Markov Decision Process (POMDP), and Deep RL models with their respective characteristics.

3.1. Reinforcement Learning and Markov Decision Process

Reinforcement learning is an experience-driven technique of machine learning where a learner (autonomous agent) makes an experience by a trial-and-error process in order to improve its choices in the future. In such a situation, the problem to solve is formulated as a discrete-time stochastic control process, so-called MDP [10]. Typically, a finite MDP is defined by a tuple (S, A, p, r) where S is a finite state space, A is a finite action space for each state $s \in S$, p is the state-transition probability from state s to state $s' \in S$ taking an action $a \in A$, and $r \subset \mathbb{R}$ is the immediate reward value obtained after an action a is performed. The agent's primary goal is to interact with its environment, at each time step $t = 0, 1, 2, ...$, to find the optimal policy π^* in order to reach the goal while maximizing the cumulative rewards in the long run. A policy π is a function that represents the strategy used by the RL agent. It takes the state s as input and returns an action a to be taken. The policy can be also stochastic. In such a case, it returns a probability distribution over all actions instead of a single action a. Mathematically, the objective of the agent is to maximize the expected discounted return R_t as follows:

$$R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor. It controls the importance of future rewards (weight) compared to the immediate one. The larger γ is, the more the estimated future rewards will be concerned. If $\gamma = 0$ the agent is "*myopic*" in being concerned only with maximizing immediate rewards.

In many applications, the agent does not have all information about the current state of the environment and had only a partial observation. Thus, the POMDP, a variant of the MDP, can be used to make decisions based on the potentially stochastic observation it receives. A POMDP can be defined by the tuple (S, A, p, Ω, O, r) where S, A, p, r are the states, actions, transitions and rewards as in MDP, Ω is a finite set of observations, and O is the observation probability from state $s \in S$ to state $s' \in S$ taking an action $a \in A$. A set of a probability distribution over the states S is maintained as "*belief state*", and the probability of being in a particular state is denoted as $b(s)$. Based on its belief, the agent selects an action $a \in A$, and move to a new state $s' \in S$ and receive an immediate reward r and a current observation $o \in O$. Then the belief about the new state is updated as follows [31]:

$$b_a^o(s') = \frac{p(o|s') \sum_s p(s'|s, a) b(s)}{\sum_{s'',s} p(o|s'') p(s''|s, a) b(s)} \tag{2}$$

In RL, the problems to resolve are formulated as an MDP. Several RL solutions can be applied, depending on the problem type considered. RL methods can be classified into multiple types as illustrated in Figure 1.

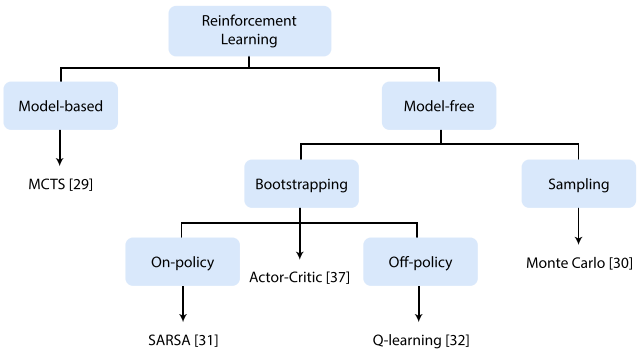


Figure 1: Classification of popular RL algorithms based on their operating features.

3.1.1. Model-based vs Model-free

With a model-based strategy, the agent learns the environment model, consisting of knowledge of state transitions and reward function. Then, simple information about the state's values is sufficient to derive policy. Whereas with a model-free, the agent learns directly from experience by collecting the reward from the environment and then updating their value function estimation. Model-based RL tends to emphasize planning to take action given a specific state, without the need for environmental information or interaction. Nevertheless, this solution fails when the state space becomes too large. Besides, these types of algorithms are very memory-intensive since transitions between states are explicitly stored. However, many model-based methods have been studied in the literature, such as Imagination-Augmented Agents (I2A) [32], Model-Based RL with Model-Free Fine-Tuning (MBMF) [33], AlphaZero [34], and Monte

Carlo Tree Search (MCTS) [35] remaining one of the frequently used methods in learning agents.

MCTS is based on a random sampling of the search space to build up the tree in memory and improve the estimation accuracy of the following choices. An MCTS strategy consists of four repeated steps: (i) selection step, where the algorithm traverses the current tree from the root node downwards to select a leaf node; (ii) expansion step, where one or more new child nodes (i.e., states) are added from the leaf node reached during the selection step; (iii) a simulation is performed from the selected node or one of the newly added child node with actions selected by the followed strategy; (iv) Backup step, where the return generated by the simulated episode is propagated back up to the root node and updates the node action values accordingly.

3.1.2. Bootstrapping vs Sampling

If the estimation of the state's values is based on another learned estimation function (i.e., a part or all successor states), the RL method is called a bootstrapping method. Otherwise, the RL method is called a sampling method whose estimation for each state is independent. This method is based only on the real observation of the states, but it can suffer from high variance, which requires more samples before the estimates reach the optimal solution. The bootstrapping method can reduce the variance of an estimator without the need to store each element.

3.1.3. On-policy vs Off-policy

An on-policy method estimates the value of the policy used to make decisions. In an off-policy method, the agent selects a different policy, called "behavior policy", regardless of the policy followed, called "target policy". State–Action–Reward–State–Action (SARSA) [37], an on-policy method, and Q-learning [38], an off-policy method, are the most RL techniques used in the literature. The equations below represent the updated function for SARSA and Q-learning respectively:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{3}$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{4}$$

where $Q(s_t, a_t)$ represents the estimate Q-value of taking action a in state s at time t; r_{t+1} represents the immediate reward returned at time $t+1$; $\max_a Q(s_{t+1}, a_{t+1})$ represents the estimate Q-value of taking the optimal action a in state s at time $t+1$; $0 \leqslant \alpha \leqslant 1$ is the learning rate; and $0 \leqslant \gamma \leqslant 1$ is the discount factor. The results of these equations are stored in a policy table, called Q-table, where rows represent the possible states, columns represent the potential actions, and cells represent the expected total reward, as depicted in Figure 2a. In SARSA, the estimation of the Q-value uses the current policy based on the same action, while Q-learning selects a new greedy action during the learning, independently of the policy being followed. This makes SARSA more cautious as it tries to take into account any negative circumstances of explorations, while Q-learning will ignore them since it followed another independent policy.

Besides the value-based algorithms, such as SARSA and Q-learning, which select actions according to the value function, policy-based methods [39] such as REINFORCE [40], Trust Region Policy Optimization (TRPO) [41], and Proximal Policy Optimization (PPO) [42] search directly for the optimal policy maximizing the expected cumulative reward. The advantage of each method depends on the task. Policy-based methods are better for continuous and stochastic environments, while value-based learning methods are more sample efficient and stable. The actor-critic methods [43] merge both: the "*critic*" estimates the value function to evaluate the action performed using the temporal difference error and the "*actor*" updates the policy distribution according to the critic suggestion. It takes advantage of both policy and value functions. The policy *actor* computes continuous actions without the need for an optimization procedure, while the value *critic* supplies the actor with a low-variance knowledge of the performance [44].

3.2. Deep Reinforcement Learning

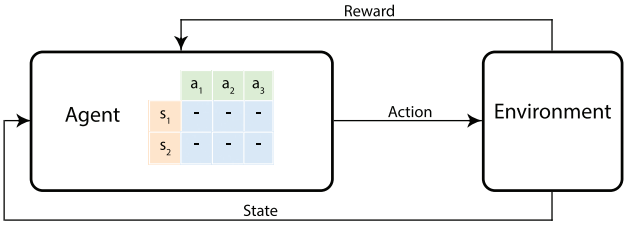
RL is effective in a variety of applications, where state and action spaces are limited. However, these spaces are usually large and continuous in real-world applications, and traditional RL methods cannot find the optimal value functions or policy functions for all states within an acceptable delay. Thus, DRL was developed to handle high-dimensional environments [45] based on the Deep Neural Network (DNN) value-function approximation as shown in Figure 2.

The DNN is a deeper version of the ANN family with, usually, more than two hidden layers [46]. Generally, an ANN consists of three principal layers, as presented in Figure 2b:

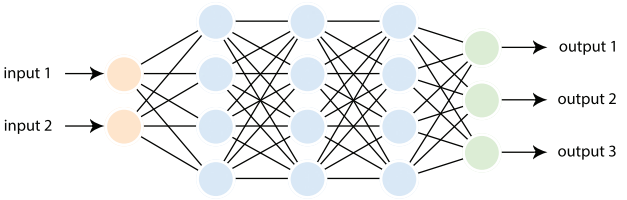
- A single input layer receives the data. The input data in DRL represent the information that describes the actual state of the environment.
- A single output layer generates the predicted values. In some works, the nodes in the output layer present the possible actions that the DRL agent can select.
- One or multiple hidden layers located between the input and output layers according to the problem complexity.

Mnih *et al.* [47] introduce in 2015 the concept of Deep Q-Network (DQN) that performed well on 49 Atari games. DQN exploits a Convolutional Neural Network (CNN), commonly applied in image processing, instead of a Q-table, as shown in Figure 2c, to analyze input images and derive an approximate action-value $Q(s, a|\theta)$ by minimizing the loss function $L(\theta)$ defined as follows:

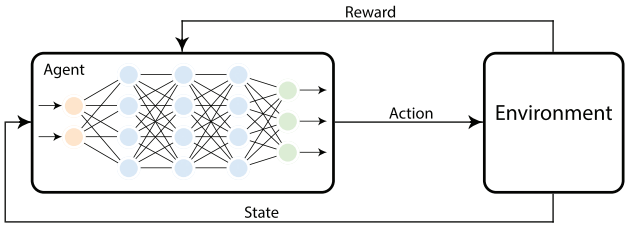
$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a'|\theta') - Q(s, a|\theta)\right)^2\right] \tag{5}$$



(a) Standard Reinforcement Learning technique.



(b) A deep Neural Network with three hidden layers.



(c) Deep Reinforcement Learning technique.

Figure 2: Principles of RL, DNN and DRL techniques.

where $\mathbb{E}[.]$ denotes the expectation function, θ and θ' represent the parameters of the predict and the target network respectively. The network is trained with a variant of the Q-learning algorithm, using stochastic gradient descent to update the weights [48]. The Q-learning in this method aims to directly approximate the optimal action-value $Q^*(s,a) \approx Q(s,a|\theta)$.

A DRL algorithm extracts features from the environment using DL while the RL agent optimizes itself by trial and error. The DRL has been extended by incorporating other ML techniques to deal with various types of problems. Transfer learning [49] aims at transferring the learned knowledge representation between different but related domains. It improves the learning process of the target agent and allows to achieve a near-optimal performance with fewer steps. Multi-task learning [50] is about learning a single policy for a group of different related tasks. It is a particular type of transfer learning, where the source and the target tasks are considered as the same. The interrelation between tasks makes the agent learning them simultaneously in order to improve the generalization performance of all the tasks. The idea behind meta-learning, also know as learning to learn, is to rapidly learn new tasks. To achieve this goal, the agent trained through various learning algorithms. Model-Agnostic Meta-Learning (MAMR) proposed by fin *et al.* [51] is a meta-learning trained by gradient descent strategy and which aims to optimize the model weights for a given neural network. To deal with some real-world scaling issues, where the agent needs many steps to get the optimal policy and achieve the goal, Hierarchical RL (HRL) forms several sub-policies that work together into hierarchically dependent sub-goals. Unlike the RL, the action space in the HRL grouped to form higher-level macro actions and then executed hierarchically rather than at a single level.

4. RL and DRL algorithms for IoT applications

Multiple research work adopted RL and DRL techniques to enhance IoT systems operations or resolve some of their issues. We categorize them into seven classes according to the addressed IoT problems: routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and edge caching. Tables 2 and 3 summarize these papers with an emphasis on RL and DRL models, their state spaces, action spaces and reward functions respectively.

4.1. Routing

IoT undergoes an exponential evolution in recent years [52, 53]. Recent developments in the technologies of wireless communications have enabled the emerging of several types of networks such as mobile networks, Vehicular Ad-Hoc Network, sensor networks. The routing functionality in these networks is a fundamental network task due to the large amount of data generated. Routing optimization with respect to the traffic demands comes up against the uncertainty of the traffic conditions. Besides, optimizing the routing configurations against a previously observed set of network scenarios or a range of feasible traffic scenarios may fail in the face of the actual traffic conditions, which are very heterogeneous. ML techniques, in particular, reinforcement learning, have been applied in several works to cope with this type of problem.

4.1.1. Path selection

For many applications, the data must be delivered to the destination within a limited period of time after their acquisition by the sensor, otherwise, it would become unusable and uninteresting. Path selection method based on Quality of Service (QoS) adapts the network routing traffic by processing packets differently according to a set of attributes such as security, bandwidth, delay (latency), and packet loss. The development of a routing protocol that ensures a balance between power consumption and data quality is a challenge due to the distributed and dynamic topology of WSNs, in addition to their limited resources.

In [54] a link quality monitoring scheme for Routing Protocol for Low-Power and Lossy Networks (RPL) has been proposed to maintain up-to-date information about networks route, and directly react to link quality variations and topology changes due to nodes' mobility. An RL technique has been applied to minimize the overhead caused by active probing operations. The proposed approach helps to improve packet loss rates and energy consumption, only for single-channel networks. To improve secure data transactions, a trusted routing scheme based on blockchain technology and RL has been introduced in [55] for WSNs. To enhance the trustworthiness of

the routing information, the proposed scheme takes advantage of the tamper-proof (i.e., detection of unauthorized access to an object), decentralization, and traceability characteristics of the blockchain, while RL helped nodes to choose a more trusted and efficient relay nodes. The RL algorithm in each routing node dynamically learns the trusted routing information on the blockchain to select reliable routing links. The results of this work prove that the integration of the RL into the blockchain system improves delay performance even with 50% of malicious nodes in the routing environment. Liu *et al.* [56, 57] have addressed the shortest path problem for intelligent vehicles. The Optimized Path Algorithm Based on Reinforcement Learning (OPABRL) proposed in these papers, used a combination of prior RL technology and searching-optimal shortest path algorithm to analyze and find a relatively shorter path with fewer turns. Compared to four algorithms commonly used in intelligent robot trajectory planning, OPABRL outperforms all methods in terms of the number of turns, the path lengths, and the running time, with a probability of 98% near to the optimal solution. The parameters such as packet loss, data content, the distance between a forward node and the sink node, and residual energy are used as metrics in [58] to design an effective security routing algorithm to ensure data security. Each node learns the behavior of its neighbors, by updating the q-value according to the collected metrics, and avoid malicious nodes in the next routing strategy. The experiments in the proposed schema were conducted under two different types of security attacks which are Black-Hole and Sink-Hole attacks. The packet delivery rate surpasses other existing trust-based methods with 2.83 to 7 times higher, depending on the percentage of malicious nodes. On the other side, the energy consumption of the proposed approach is too high compared to the others, especially with the black hole attack.

Zhao *et al.* [59] have developed a deep RL routing algorithm to improve crowd management in smart cities, called DRLS, by meeting the latency constraints of people's service demands. Compared to the classic link-state protocols, Open Shortest Path First (OSPF) and Enhanced-OSPF (EOSPF), DRLS performance outperforms in terms of service access delay and successful service access rate, with more stable management of network resources.

4.1.2. Routing based on energy efficiency

Low-power IoT networks rely on constrained devices, and they are battery-powered. In this type of networks, packet transmission is hop by hop, and the packets cross multiple intermediate nodes to reach the base station. The energy consumption of the closest nodes to the sink is higher because they serve as relay nodes. Since energy is mostly consumed by the communication module, several routing approaches have been proposed to tackle this problem in order to increase network lifetime.

The authors in [60, 61] have tried to maximize the network lifetime of WSNs by improving routing strategies using RL. The proposed methods mainly consider the link distance, the residual energy of the node and the neighbors, in the definition of reward function, to select the best paths. We have to note that [60] implements a flat architecture, which makes the proposed solution only intended for small networks with low requirements and not suitable for large scale WSNs. A multi-objective optimization function has been proposed for WSNs data routing in [62], using clustering and the RL method SARSA, defined as Clustering SARSA (C-SARSA). Fair distribution of energy and maximize vacation time of the Wireless Portable Charging Device (WPCD) are the two main objectives. For that, the data-generating rate, energy consumption rate of receiving/sending the data, and the time of arrival, charging, traveling, and field have been considered. According to the residual energy, each node determines its willingness, which reflects if it can participate in the determination of the data route or not.

The routing problem in UnderWater Sensor Networks has been studied by many works due to the particularity of this type of network. The topology in an underwater network is more dynamic compared to WSN, where each node is independent and frequently changes its position relative to other sensors with the water currents. This causes instability of communication links, reduces their efficiency, and increases energy consumption [63]. Several routing challenges face UWSN, such as high propagation delay, localization, clock synchronization, and radio waves attenuation, especially in salt-water.

In [64, 65], a Q-learning algorithm has been proposed to ensure data forwarding in UWSN. These approaches consider the propagation delay to the sink and the energy consumption of the sensor nodes to learn the best path. Li *et al.* [66] have proposed a novel routing protocol based on the MARL protocol for Underwater Optical Wireless Sensor Network (UOWSN). The MARL agent determines the next neighbor according to the link quality, to reduce the latency communication and the residual energy of the node, and maximize the network lifetime.

Kwon *et al.* [67] have formulated a distributed decision-making process in multi-hop wireless ad-hoc networks using a Double Deep Q-Network [68, 69]. Double DQN handles the problem of overestimating Q-values by selecting the best action to take according to an online network, and calculate the target Q value of taking that action using a target network. Each relay node adjusts its transmission power to increase or decrease the wave range in a way to improve both network throughput and the amount of the corresponding transmission power consumption.

Path selection, in terms of QoS and distance, as well as the routing strategies based on energy-efficient, are the two most routing optimization problems that have been addressed by the RL techniques in recent years.

4.2. Scheduling

Due to the heterogeneous and dynamic nature of IoT network infrastructure, scheduling decisions became a fundamental problem in the development of IoT systems. Briefly, the scheduling problem is to determine the sequence in which operations will be executed at each control step or state. Smart and self-configuration devices should be used to adapt schedule decisions based on environmental changes, and satisfying certain restrictions, such as hardware resources and application performance. In order to improve the trade-off between energy

consumption and the QoS, RL techniques have been applied in many works to adapt tasks, data transmission, and time slot scheduling.

4.2.1. Task scheduling

A task scheduling process assigns each sub-task of different tasks to a selected and required set of resources to support user goals. Different studies have applied RL techniques to optimize task scheduling process in IoT networks.

A cooperative RL among sensor nodes for task scheduling has been proposed in [70, 71], to optimize the tradeoff between energy and application performance in WSNs. Each node takes into consideration the local state observations of its neighbors, and shares knowledge with other agents to perform a better learning process and gives better results than a single agent. Wei *et al.* [72] have combined the Q-learning RL method with an improved supervised learning model Support Vector Machine (ISVM-Q), for task scheduling in WSN nodes. The ISVM model takes as input the state-action pair and computes an estimation of the Q value. Based on this estimation, the RL agent selects the optimal task to realize. The experimental results show that the proposed ISVM approach improves application performance while preserving network energy by putting the sensor and communication modules to sleep mode when necessary. These results remain valid even with a higher number of trigger events and with different learning rates and discount factors values.

4.2.2. Data transmission scheduling

The wireless sensor nodes usually have limited available power. Thus many research studies attempt to optimize data transmission of collected measurements in order to extend network lifetime and ensure stability.

Kosunalp *et al.* [73] addressed the problem of data transmission scheduling by extending an ALOHA-Q strategy [74] to be integrated into the Media Access Control (MAC) protocols design. ALOHA-Q uses slotted-ALOHA as the baseline protocol for access channel with a benefit of simplicity. The Q-value of each slot into the time frames represents the willingness of this slot for reservation. The simulation result shows that ALOHA-Q outperforms existing scheduling solutions and it can provide better throughput and is more robust with an additional dynamic ϵ-greedy policy.

The authors in [75] have addressed the transmission scheduling optimization in a maritime communications network based on Software Defined Network (SDN) architecture. A deep Q-learning approach combined with a softmax classifier (S-DQN) has been implemented to replace the traditional algorithm in the SDN controller. The DQN was used to establish a mapping relationship between the received information and the optimal strategy. S-DQN aims to optimize scheduling in a heterogeneous network with a large volume of data to manage. Yang et Xie [76] have attempted to solve the transmission scheduling problem in Cognitive IoT (CIoT) systems with high dimension state spaces. For that, an actor-critic DRL approach based on a Fuzzy Normalized Radial Basis Function neural network (AC-FNBRF) has been proposed to approximate the action function of the actor and the value function of the critic. The performance of the proposed approach has been compared with classical actor-critic RL, deep Q-learning, and greedy policy RL algorithm. Simulations show that the proposed AC-FNBRF outperforms the others with a gain that reaches 25% on power consumption and 35% on transmission delay when the packet arrival rates are high.

4.2.3. Time slot scheduling

Time slot scheduling algorithms dynamically allocate a unit of time for a set of devices to communicate, collect or transfer data to another device in order to improve throughput and minimize the total latency. RL and DRL methods are leveraged in many research work to provide better scheduling.

Lu *et al.* [77] have integrated the Q-learning technique into the exploration process of an adaptive data aggregation slot scheduling. The RL approach converges to the near-optimal solution, and the nodes have the capability to reach the active/sleep sequence, which increases the probability of data transmission and saves sensor energy. However, compared to three exiting methods, namely Distributed Self-learning Scheduling, Nearly Constant Approximation, and Distributed delay-efficient data Aggregation Scheduling, the performance of the proposed RL approach does not exceed all of them in terms of average delays and residual energy.

A deep Q-learning has been applied in [78] for scheduling in a vehicular network, to improve the QoS levels and promote the environment safety without exhausting vehicular batteries. The RL agents have been implemented in centralized intelligent transportation system servers, and learn to meet multiple objectives, such as reducing the latency of safety messages and satisfying the download requirements for vehicles before leaving the road. The performance of the proposed DQN algorithm exceeds several existing scheduling benchmarks in terms of vehicles' completed request percentage (10% – 25%) and mean request delay (10% – 15%). In terms of network lifetime, DQN outperforms all its competitors except the Greedy Power Conservation (GPC) method in some situations with large file sizes or when the density of the vehicular network increases.

The performance of the proposed RL and DRL based approaches has been studied in task, data transmission, and time slot scheduling.

4.3. Resource Allocation

As the number of connected objects increases, a large volume of data is generated by IoT environments. This explosion is due to the various IoT applications developed to improve our daily life, such as smart health, smart transportation, and smart city. Therefore, to be able to adapt to the environment change and cope with the specific requirement of applications, like reliability, security, real-time, and priority, it is necessary to rely on a Resource Allocation (RA) process. The goal is to find the optimal resources allocation, to a given number of activities, in order to maximize the total return or minimize the total costs.

Xu *et al.* [79] have addressed the RA problem to maximize the lifetime of WBAN using RL. The harvested energy, trans-

mission mode and power, allocated time slots, and relay selection are taken into account to make the optimal decision. The authors in [80] have proposed an extensible model for Self-Organizing MAC (SOMAC) for wireless networks, to switch between the available MAC protocols (i.e., CSMA/CA and TDMA) dynamically using RL and improve the network performance according to any metric, such as delay, throughput, packet drop rate, or even a combination of those, chosen by the network administrator.

In [81], a DRL based vehicle selection algorithm has been designed to maximize spatial-temporal coverage in mobile crowdsensing systems. A DRL resource allocation frameworks for Mobile Edge Computing (MEC) have been proposed in [82, 83]. The authors have used different DRL techniques and metrics in their modeling of these systems. For instance, [83] has applied a Monte Carlo Tree Search method based on a multi-task RL algorithm. The work has improved the traditional DNN, by splitting the last layers, to build a sublayer neural network for high-dimensional actions. The service latency performance was significantly better in the random walk scenario and the vehicle driving-based base station switching scenario, compared to the deep Q-network with percentages reaching 59% in some cases.

RL approaches, including DRL, have been adopted for resource allocation in wireless networks, internet of vehicles, and mobile edge networks.

4.4. Dynamic Spectrum Access

With the emergence of the IoT paradigm and the growing number of devices connecting and disconnecting from the network, it was necessary to develop new dynamic spectrum access solutions. The DSA is a policy that specifies how equipment can efficiently and economically share available frequencies while avoiding interference between users.

In [84], two Q-learning algorithms have been integrated into the channel selection strategy for Industrial-IoT, to determine which channels are vacant and of good quality. Both proposed RL algorithms aim to assess the future occupation of the channels and to sort and classify the candidate channel list according to the channel quality. Compared to five spectrum handoff schemes, the results showed a remarkable improvement in latency and throughput performance under diverse IIoT scenarios. A Non-Cooperative Fuzzy Game (NC-FG) framework has been adopted in [85] to address the requirement for an optimal spectrum access scheme in stochastic 5th-Generation wireless systems (5G) WSNs. To reach the Nash equilibrium solution of the NC-FG, a fuzzy-logic inspired RL algorithm has been proposed to define the robust spectrum sharing decision and adjust the channel selection probabilities accordingly. We have to note that the values of the various parameters of the implemented RL system are missing in the paper.

The problem of dynamic access control in 5G cellular networks has been studied also by Pacheco-Paramo *et al.* [86]. The authors proposed a real-time configuration selection scheme based on the DRL mechanism to dynamically adjust the access class barring rate according to the changing traffic conditions and minimize collision cases. The training results show that the proposed solution is able to reach a 100% success access probability for both Human-to-Human and Machine-to-Machine user equipment and with a low number of transmissions. To achieve this level of performance, the training of the proposed mechanism is almost three times longer than the Q-Learning based solution. In wireless networks, Wang *et al.* [87] have implemented a DRL-based channel selection to find the policy that maximizes the expected long-term number of successful transmissions. The DSA problem has been modeled as a POMDP in an unknown dynamic environment. At each time slot, a single user selects a channel to transmit packet and receive a reward value based on the success/failure status of transmission. The authors designed an algorithm that allows the DQN to re-train a new good policy, only if the returned reward value is reduced by a given threshold. This can degrade the performance of the proposed solution, especially with dynamic IoT environments.

Research papers that employ RL and Deep RL to solve dynamic spectrum access problems focus mainly on networks where communication traffic is changing and unknown.

4.5. Energy

For battery-powered devices in IoT systems, optimizing energy consumption and improving network lifetime are fundamental challenges. The difficulty of replacing or recharging the nodes, especially in unreachable areas, motivated researchers to employ the RL technique to find a better compromise between residual energy and application constraints.

In [88] a Transmission Power Control (QL-TPC) approach has been proposed to adapt, by learning, the transmission power values in different conditions. Every RL agent is a player in a common interest game. This game theory provides a unique outcome and leads to a global benefit by minimizing transmission power with a percentage of packet reception ratio always higher than 95%. Based on traffic predictions and transmission state of neighbor nodes, an RL agent has been designed in [89] to adapt between the sleep-active node duty-cycle, by adjusting the MAC parameters and reduce the number of slots in which the radio is on. The drawback of the proposed solution is that it has a high probability of packet loss. Soni et Shrivastava [90] have picked up nodes in cluster sets using Q-learning and then collect data from cluster heads using a mobile sink. This solution has a double advantage: first, clustering saves the energy consumption of the nodes by reducing the number of hops and distance to the cluster head. The second advantage is that the mobile sink visits only the interested cluster head which sends a request to the sink for data collection.

Energy Harvesting (EH) is an alternate process to provide some energy to the nodes by deriving power from external sources (e.g., solar, thermal, wind) and extend the lifetime of the sensor network. Due to the stochastic behavior of this technology, since most of these energy sources vary over time, using EH brings new challenges to energy management of how to maximize the harvested power and energy use efficiency. Several energy management schemes using RL have been proposed.

In [91] an RL energy management (RLMAN) has been proposed using an actor-critic algorithm to select the appropriate

throughput based on the state of charge of the energy storage device. The problem of energy management is presented as a Cooperative Reinforcement Learning (CRL) in [92], where agents share information to regulate active/sleep duty cycle. In this system, each EH-node seeks to keep both its node and the next hop node alive.

Deep RL approaches have been employed to improve nodes' performances, for the large IoT energy harvested networks In [93] an end-to-end approach has been proposed to control IoT nodes and select the value or define the interval of the duty cycles. For this, a PPO policy gradient method using neural networks as function approximators has been applied, giving better results compared to SARSA. Sharma *et al.* [94] have exploited the mean-field game combined with multi-agent Q-learning to find out the optimal power control and maximize the obtained throughput. In simulations, the authors have only focused on sum-throughput to show that the proposed approach can achieve performance close to DNN-based centralized policies, without requiring information on the state of all the nodes in the network.

Energy optimization has been and will be continually an interesting research topic for wireless IoT networks. The existing work have shown that the application of RL and DRL approaches allowed to extend node's lifetime and improve harvested networks.

4.6. Mobility

Mobile nodes in WSNs, such as a robot or mobile sink, allows overcoming the various trade-offs related to the characteristic of these networks. To cope with the energy issue, for example, sink mobility has been exploited in many systems to extend network lifetime and solve the hotspot problem (also known as energy hole-problem), as depicted in Figure 3.

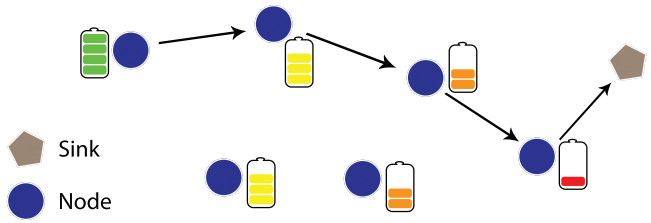


Figure 3: Illustration of the energy hole-problem.

As we have mentioned previously, several IoT devices rely on low-cost hardware, and they are not equipped with location sensors (e.g., LPS, GPS, or iBeacon). Therefore, an intelligent system must be integrated into mobile nodes in order to find the optimal trajectory to follow. With the trial-and-error strategy followed by the RL agents, the mobile nodes have the capacity to explore the network environment, make internal decisions to find the right trajectory, and adapt to dynamic networks.

Wang *et al.* [95] have addressed the problem caused by scaling-up in WSNs and propose a location update scheme of the mobile sink node to achieve more efficient network topology. First, the sink node updates its location by collecting information from certain key nodes and searches for the best performing location to define it as the final location. Then, a Window-based Directional SARSA(λ) algorithm (WDS) is designed to build an efficient path-finding strategy for the sink node. Through two simulation scenarios, a simple path, and a longer path with traps, the WDS algorithm is always able to find the optimal route to the sink with only a 48% to fall into a trap.

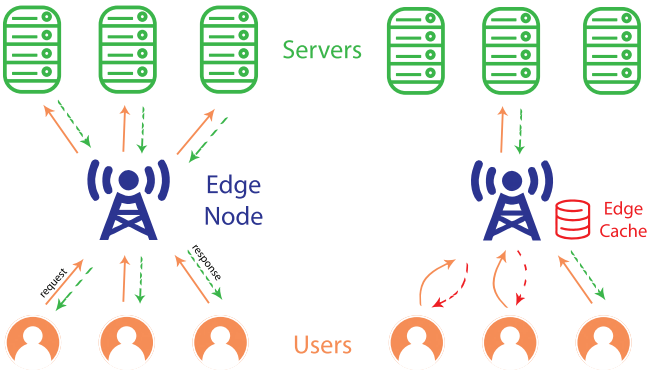
The authors in [96] have developed an indoor user localization system based on BLE for smart city applications. Their solution extends the DRL to semi-supervised learning that utilizes both labeled and unlabeled data. The model collects the Received Signal Strength Indicator (RSSI) from a set of fixed Bluetooth devices with a known position to provide the current location and the distance to the target point. Simulations show an improvement that can reach 23% in target distance and at least 67% more rewards compared to the supervised DRL scheme. Liu *et al.* [97] have proposed an Ethereum blockchain-enabled data sharing scheme combined with DRL to create a safe and reliable IIoT environment. A distributed DRL based approach was integrated into each mobile terminals to move to a location and achieve the maximum amount of collected data. Blockchain technology was used to prevent attacks and network communication failure while sharing the collected data. Simulation results showed that compared to a random solution, the DRL algorithm can increase the geographical fairness ratio by 34.5%. The problem of data collection in WSNs has been addressed also in [98]. The authors propose a single mobile agent based on DRL to learn the optimal route path while improving data delivery to the sink and reduce energy consumption. The proposed method employs a DNN combined with the actor-critic architecture, where it takes as input the state of the WSN, defined by the locations of each node in the environment, and outputs the traveling path.

The employment of RL approaches, especially the Deep RL, have allowed to manage the network topology and collect data via mobile nodes.

4.7. Caching

With the rapid increase in IoT devices and the number of services over mobile networks, the amount of wireless data traffic generated is continuously increasing. However, the limit of link capacity, the long communication distance, and the high workload introduced in the network pose significant challenges to satisfy the Quality of Experience (QoE) or the QoS required by applications. Edge caching is a promising technology to tackle these network problems. The goal of caching is to reduce unnecessary end-to-end communications by keeping popular content at edge nodes close to users. Thus, the requested data can be obtained quickly from nearby edge nodes, which reduces the redundant network traffic and meet the low-latency requirement. Compared to the network in Figure 4a, the edge cache in Figure 4b allowed to reduce the number of requests required toward remote servers.

In [99], a DRL based caching policy has been proposed to solve the cache replacement problem, for IoT content, with limited cache size. Taken into consideration both the fetching cost



(a) Without edge cache. (b) With edge cache.

Figure 4: Illustration of the difference between standard and cache-enabled edge scenarios in IoT networks.

and the data freshness, the developed framework can make efficient decisions without assuming the data popularity or the user request pattern. The DRL based caching policy has been compared to the two caching policies Least Recently Used (LRU) and Least Fresh First (LFF). The simulation results demonstrate that the proposed policy achieves better performance in terms of cache hit ratio, data freshness, and data fetching cost with different configurations. An actor-critic DRL is applied in [100] to realize a joint optimization of the content caching, computation offloading, and resource allocation problems in fog-enabled IoT networks. Two DNNs have been employed to estimate the value function with a large state and action space in order to minimize the average transmission latency. Evaluation results show the effect of radio resource optimization and caching on decreasing end-to-end service latency. The proposed approach outperforms the performance of offloading all tasks at the edge when the computational capability increase, but it starts to degrade when the computational burden becomes heavy.

Time-varying feature, as caching issues, has been solved using deep reinforcement learning techniques.

5. Discussion and Lessons Learned

Tables 2 and 3 summarize the research in each wireless IoT issue. For each of surveyed paper, we identified the used RL and DRL models, their state spaces, action spaces, and reward functions.

Through this survey, we found that both RL and DRL algorithms are able to enhance network performance, such as lower transmission power [67, 79, 88], provide better routing decision-making as the works described in the section 4.1, and higher throughput [76, 91, 92], and these in various wireless networks, including underwater wireless sensor network [64, 65, 66], internet of vehicles [56, 78, 81], cellular network [85, 86], Wireless Body Area Networks [79], etc. Reinforcement learning models allow a wireless node to take as input its local observable environment and subsequently learn effectively from its collected data to prioritize the right experience and choose the best next decision. The Deep RL approach, a mix between RL and DL, allows for making better decisions when facing high-dimensional problems and to solve scalability issues by using neural networks, one of the challenges, for example, in edge caching methods.

In terms of the time complexity of the proposed RL solutions, [70] proposes a constant time approach since the size of the set of tasks and the number of the neighbor nodes are fixed at initialization. But most of the other proposed algorithms, such as [54, 72, 79, 95, 101], require a quadratic time complexity with an execution time that increases exponentially as the size of the state and/or the action space increases. The authors in [76] try to decrease the Computational complexity by combining the hidden layer nodes which have similar functions. The improved deep Q-network, S-DQN [75], reduces the computation and time complexity by more than 90% to reach Q-value converge.

The cost in terms of energy consumption has been evaluated in many surveyed works. The proposed routing protocol in [60] makes a significant improvement in three situations: the first node dies, the first node isolates to the sink, and the time until the network cannot accomplish any packet delivery. With small and large scale network scenarios, the MAC protocol designed in [89] allows to extend nodes lifetime, up to 26 times, by reducing the average activated slots. The authors obtained a percentage of Packet Delivery Ratio (PDR) near that with a duty cycle at 100%. The routing method in [58] focuses more on ensuring a higher PDR when the network is facing attacks, however, it introduces high energy consumption. The RL method in [64] consumes less energy among the compared data forwarding methods in different network sizes with a percentage reaching 37.26%. This solution acquires a better value of information but achieves a PDR slightly lower. The power consumption by the actor-critic system in [76] increases with the packet arrival rate. That always remains lower than the state of art algorithms, even during the learning process. Another actor-critic solution has been proposed in [98] shows that DRL can reduce the consumed energy by the mobile agents over time as the training process progresses. The deep MCTS method applied in [83] outperforms all methods in terms of energy consumption, with different computing capability of edge servers and a varied number of mobile devices ranging from 10 to 260, while always ensuring a minimal average service latency.

The goal of an RL agent is to learn the policy which maximizes the reward it obtains in the future. A Single-Agent Reinforcement Learning (SARL) approach is based on using only one agent in a defined environment that makes all the decisions. In the IoT network, the SARL can be deployed in the base stations or monitoring nodes (e.g., sink node), and the RL agent interacts according to the state of the whole network. The network can also have several agents deployed in one or multiple nodes but each focuses on optimizing only its own environment, for example, battery level, the channels available for this node, the neighbor nodes. When adopting more than one agent that interacts with each other and their environment, the system is called MARL. The MARL accelerate the training in distributed solutions that can be efficient than centralized SARL since each RL agent combines both its own experience and that of the other

Table 2: A summary of applied RL methods and models with their associated objectives.

Obj.	Ref.	RL Method	State	Action	Reward	Network
Routing	[54]	Multi-armed bandit	Link quality	Selects the set to probe	Trends in link quality variations	WSN with mobile nodes
	[55]	SARSA	Current position of packets (i.e., routing node)	Select a routing node	Delivered tokens	WSN
	[56, 57]	Q-learning	Grid map: obstacle or non-obstacle	Select a path	Related to path length and number of turns	IoV
	[58]	Q-learning	Behaviors of neighboring nodes	Select neighbor node	Forwarded data packet	WSN
	[60]	Custom Q-value	Current position of packets (i.e., sensor node)	Select a neighboring node	Related to neighboring residual energy, distance to the neighbor, and hop count between neighbor node and sink	WSN
	[61]	–	–	Select a route	–	WSN
	[62]	SARSA	The ratio of the remaining energy to the drain rate of the energy	Select a forwarding ratio of route request packet	Energy drain rate	WPCD
	[64]	Q-learning	Transmission status in previous time slot	Select a node	Related to information timeliness and residual energy	UOWSN with passive mobility
	[65]	Q-learning	Position of the sensor node	Select an accessible node	Transmission distance	UWSN
	[66]	Q-learning (Modified)	{Busy, Idle}	Select neighbors of each node	Residual energy and link quality	UOWSN
Scheduling	[70]	Q-learning	[State of sensing area, state of transmitting queue, state of receiving queue]	{Sleep, Track, Transmit, Receive, Process}	Task completion success	WSN
	[71]	SARSA (Modified)	{Idle, Awareness, Tracking}	{Detect Targets, Track Targets, Send Message, Predict Trajectory, Intersect Trajectory, Goto Sleep}	Related to residual energy, maximum energy level, and number of field-of-view detected target's positions	WSN
	[72]	Q-learning	[Sensing area, Transmitting queue, Receiving queue]	{Sleep, Sense, Send, Receive, Aggregate}	Related to scheduling energy consumption and applicability predicates	WSN
	[73]	Q-learning	List of nodes	Select a slot	Transmission success	WSN
	[77]	Q-learning	Selected active slots in the previous frame	Select an active slot for the current frame	Transmission and acknowledgment success	WSN
RA	[79]	Q-learning	Data and energy queue lengths	Select a transmission mode, a time slot allocation, a relay selection, and a power allocation	Related to transmission rate and consumed transmission power	EH-WBAN
	[80]	Q-learning	Available MAC protocols	Select a MAC protocol	Related to the pourcentage gain of a network performance metric	Wireless Network
DSA	[84]	Q-learning	Bandwidths: occupied or unoccupied	Select a set of channels	Probability of sensing a vacant channel	IIoT
	[85]	–	Spectrum sharing decision	Adjust channel selection probabilities	–	5G WSN

– Not mentioned

(Continued on next page)

Table 2: A summary of applied RL methods and models with their associated objectives *(Continued)*.

Obj.	Ref.	RL Method	State	Action	Reward	Network
Energy	[88]	Q-learning	Link status	Select a transmission power levels	Related to Packet Reception Ratio and Power levels	WSN
	[89]	Q-learning	Slot information of the current node and its neighborhood	Active or sleep mode	Related to the amount of successfully transmitted, received, and overheard received packets	WSN
	[90]	Q-learning	Neighbouring cluster head which can be selected	Select a neighbouring node	Link cost	WSN with mobile sink
	[91]	Actor-Critic	[Residual energy required to operate, Energy storage capacity]	Select the throughput F_{min} and F_{max}	Related to normalized residual energy and throughput	EH-WSN
	[92]	Q-learning	[Residual energy, Throughput]	[Stay sleep, Turn on collection, Turn on processing, Turn on transmission]	Takes different values for each possible outcome	EH-WSN
Mobility	[95]	SARSA (Modified)	Location of the sink	{Turn left, Turn right, Forward, Backward}	+1 if final location, 0 otherwise	WSN with mobile sink

agents. In the surveyed papers, we note that only 20% of the RL based approaches use MARL, and only one paper [94] in DRL approaches. This can be explained by the fact that the MARL approach requires an efficient and regular synchronization system, which can cause overloads on IoT networks and reduce the performance and the lifetime of the nodes. In addition, Deep learning requires much more computational performance than the standard RL. This would make the deployment of such an algorithm in a distributed network with low power nodes more difficult and less efficient regarding resources consumption.

Simulation and emulation are well-established techniques to imitate the operation of a real-world process or system over time [102]. They are applied by researchers and developers during the testing and validation phases of new approaches. This is the case with machine learning technologies that require a large amount of resources and time for the training and testing phases. MATLAB [103, 104], Cooja [105], OMNET++ [106], and Network Simulator (NS-2/3) [107, 108] are the most network simulator that the authors use to evaluate the application of their RL approaches in IoT networks. To evaluate DRL based approaches, the authors turn more towards tools using python as a programming language, such as TensorFlow [109] and OpenAi-Gym [110], which offer several libraries for ML and DL. In addition to the simulation results, the authors in [54, 67, 73, 80, 88, 89] have evaluated the performance of their approaches in real-world experiments. This gives more realistic results of the proposed approaches performances for IoT networks.

The availability of a global view of the network or the collection of all the necessary information from the environment is not always guaranteed in IoT environments. Two surveyed papers [88, 87] have addressed the problems of partial observations about the overall environment by the RL agent.
The authors in [88] rely on a decentralized system where each agent is independent but simultaneously influences a common environment. Thus, a multi-agent Decentralized POMDP (Dec-POMDP) is considered in a wireless system with more than one transmitter node, where each one relies on its local information. Otherwise, each agent needs to know and keep track of action decision and reward value per transmitted packet of all other agents. To avoid network overhead and reduce complexity, such information is not exchanged between the agents. Based on stochastic games, indirect collaboration among the nodes is obtained through the application of the common interest in game theory since they aim to improve the total reward by helping each other.
In [87], the DSA problem has been formulated as a POMDP with unknown system dynamics. The problem has been formulated as follows: a wireless network is considered with multiple nodes, dynamically choosing one of N channels to sense and transmit a packet. The difficulty comes from the fact that the full state of all channels can not be all observable since the node can only sense one channel at the beginning of each time slot. However, based on the previous sensing decisions and observations, the RL agent infers a distribution on the state of the system. Considering the advantage of the model-free over the model-based in this type of problems, a Q-learning was applied by considering the belief space and converting the dynamic multi-channel access into a simple MDP. The state-space size grows exponentially as it becomes large, which requires using deep Q-learning instead of the Q values table.

It is evident that from 2016 to February 2020, most re-

Table 3: A summary of applied DRL methods and models with their associated objectives.

Obj.	Ref.	DRL Method	State	Action	Reward	Network
Routing	[59]	Deep Q-learning	Location of requests	Select a route requests	Related to request access success, resource usage balance degree, and data transmission delay	Smart city network
	[67]	Double Deep Q-Network	Number of relay nodes	Transmission range	Related to throughput improvement and transmission power consumption	Wireless ad-hoc Network
Scheduling	[75]	Deep Q-learning	[Channel state, Cache state]	Dispatch or not a data packet to a relay ship for caches state, channels state, and energy consumption	Related to signal-noise ratio, terminals' cache state, and energy consumption	Maritime Wireless Networks
	[76]	Actor-Critic	[Channel status, Channel access priority level, Channel quality, Traffic load of the selected channel]	{Transmit power consumption, Spectrum management, Transmission modulation selection}	Related to transmission rate, throughput, power consumption, and transmission delay	CIoT
	[78]	Deep Q-learning	Underlying network characteristics	Select an intelligent transportation system server or a vehicle	Sum of IoT-GWs' power consumption, waiting time of the vehicles to receive any service, delay of completed service requests, penalty for incomplete service request and early cut-off of one of the IoT-GWs	IoV
RA	[81]	Deep Q-learning	Covered times	List of selected vehicles	Cost of the sensing tasks	IoV
	[82]	Deep Q-learning	Data rate of user equipment and computation resource of vehicular edge server and fixed edge server	Determine the setting of vehicular edge server and fixed edge server	Vehicle edge computing operator's utility	Vehicle Edge Network
	[83]	Monte Carlo tree search	[Computing capability state, radio bandwidth resource state, task request state]	[Bandwidth, Offloading ratio, Computation resource]	Related to the end node in the search path	Mobile Edge Network
DSA	[86]	Deep Q-learning / Double Deep Q-learning	Received preambles success and barring rate	Select barring rate value	Avoid or reach a defined limit	5G
	[87]	Deep Q-learning with Experience Replay	Channels' state : good or bad	Select a channel	Transmission success	Wireless Network

(Continued on next page)

Table 3: A summary of applied DRL methods and models with their associated objectives *(Continued)*.

Obj.	Ref.	DRL Method	State	Action	Reward	Network
Energy	[93]	Proximal Policy Optimization	[Level of the energy buffer, Distance from energy neutrality, Harvested energy, Weather forecast of the day]	Select a duty cycles value	Distance to energy neutrality	IoT
			[Level of the energy buffer, Harvested energy, Weather forecast of the whole episode, Previous duty cycle]	Select the maximum duty cycle	Level of the energy buffer	
	[94]	Deep Q-learning	Energy arrivals and channel states to the access point of the node	Transmit energy	Sum throughput	EH network
Mobility	[96]	Deep Q-learning	[Vector of RSSI values, Current location, Distance to the target]	{West, East, North, South, NW, NE, SW, SE}	Positive if distance to the target point is less than a threshold, negative otherwise	IoT
	[97]	Actor-Critic	{[Data distribution, Location of Mobile terminal, Past trajectories]}	Moving direction and distance	Energy-efficiency	IIoT
	[98]	Actor-Critic	Coordinates of the source nodes	Mobile node movement	Negative of the consumed energy	WSN with mobile agents
Caching	[99]	Actor-Critic	Values of information about cached/arrived data items	Replace or not the data cached	Sum utility of requested data items	IoT
	[100]	Actor-Critic	[Size of input data, computation requirement, popularity of requesting, storage flag, link quality]	[Assign BSs to requesting service, requesting decision, computation tasks location, computational resource blocks]	Related to the time cost function for computation offloading requests and for content delivery requests	Fog computing network

searchers in these studies have given considerable interest to apply RL and DRL according to the requirements of the application and the target set. We note that most research focuses on applying the RL technique in routing, scheduling, and energy. While in resource allocation, mobility, and caching, researchers are more focusing on applying the DRL technique. In these types of applications, agents are located in unconstrained devices, such as at the edge router and crowdSensing servers, and which can be easily extended with more computation resources.

The majority of RL approaches are using Q-learning as a method for training their agents, and few of them are using SARSA method. This can be explained by the fact that SARSA takes into consideration the performance of the agent during the learning process, whereas with Q-learning, authors only care about learning the optimal solution towards which they will eventually move. We also note that the actor-critic method is mainly used with DRL approaches due to the difficulty of training an agent in many cases. This is due to the interaction instability, during the learning process, between the actor and critic, as one of the weaknesses of the methods based on the value function.

As shown in Figure 5, nearly a third of the research covers the routing issue. RL algorithms show good results since they are flexible, robust against node failures, and it can maintain data delivery even if the topology changes. The management of energy consumption and harvesting still represents a significant research challenge, with the main objective to extend the lifetime of the IoT network. Depending on the network characteristics and application constraints, the definition of the network lifetime differs. The three major definitions are [111]: (i) the time until the first node death; (ii) the time until the first node becomes disjoint; (iii) the time until all nodes die or failure to reach the base station. Various other definitions have been proposed and reported in the literature, where the researchers are using thresholds such as the percentage of dead nodes, packet delivery rate, remaining energy.

Some miscellaneous issues have been solved using RL techniques for wireless IoT networks, but they are not well cov-

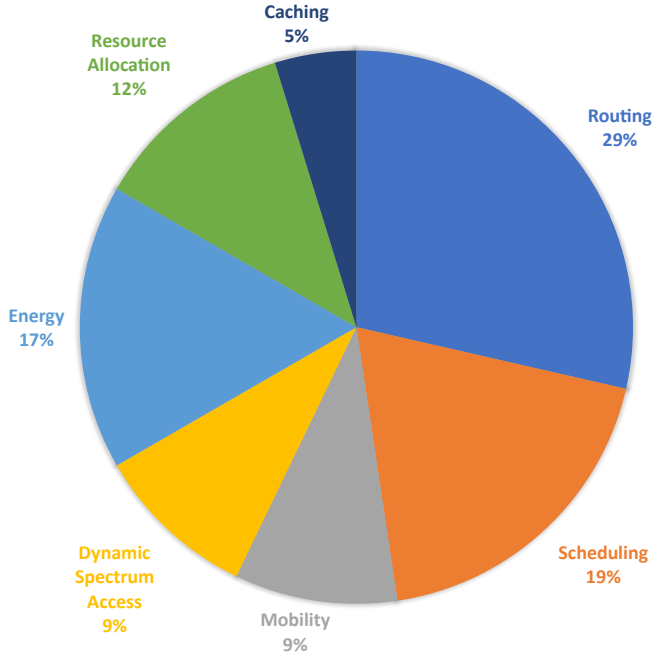


Figure 5: Percentage of research papers according to the IoT issues addressed by RL and DRL techniques.

ered during the period of the surveyed papers. In terms of IoT security, a DQN-based detection algorithm has been proposed by Liang *et al.* [101] for virtual IP watermarks, to ensure the safety of low-level hardware in the intelligent manufacturing of IoT environments and real-time detection for virtual intellectual property watermarks. For the deployment and topology control issue, Renold *et al.* [112] have developed a Multi-agent Reinforcement Learning-based Self-Configuration and Self-Optimization (MRL-SCSO) protocol for effective self-organization of unattended WSNs. To maintain a reliable topology, the neighbor with the maximum reward value is selected as the next forwarder.

6. Challenges

The different surveyed papers show that RL and DRL techniques are able to solve multiple issues in IoT environments by making them more autonomous decision-making. However, the application of RL and DRL techniques still has certain challenges:

- The identification of the RL method to be applied in a specific IoT issue can be a difficult task. We observe from the surveyed papers, that the majority of work use the Temporal-Difference [113] learning methods (e.g., SARSA, Q-learning). This poses another challenge since the performance of these methods is affected by multiple parameters such as the learning rate and the discount factor. Each of these parameters can take a real number ($\mathbb{R}$) of an infinity of value between $[0, 1]$. In DRL, the task of tuning the parameters of the applied techniques becomes more difficult since it is affected by the number and type of neural hidden layers as well as the loss function.

- The RL research studies try to identify all the possible scenarios that an RL agent may face to define the "*optimal*" reward function that achieves the goal quickly and efficiently. Sometimes agents encounter new scenarios where the defined reward function can lead to unwanted behavior. One of the solutions proposed for this problem is to recover the reward function by learning from demonstration, known as Inverse RL [114, 115, 116].

- Unlike supervised and unsupervised learning, RL et DRL methods have no separate training steps. They always learn by a trial-and-error process, as long as the agent did not reach a final state. In this case, the performance of the agent varies according to the historical data (i.e., the encountered environment states and the executed actions), and the followed exploration strategy. The offline RL [117] strategy can accelerate this process by collecting a large dataset from past interactions and train the agent for many epochs before deploying the RL or DRL model into the real environment. On top of that, the sequential steps of the trial-and-error process can extend the convergence time. The Advantage Actor-Critic (A2C) [118] and Asynchronous Advantage Actor-Critic (A3C) [119], two variants of the actor-critic algorithm, can handle this by exploring more state-action space in less time. The principal difference is that several independent agents are trained in parallel with a different copy of the RL environment and then update the global network.

- The problem of supporting network mobility remains not well studied in many surveyed works. A wireless network may include one or more mobile nodes, which makes its structure dynamic and can be relatively unstable. In fact, if the RL mechanisms do not explicitly support network mobility, dysfunction, or degradation in performance can affect the system. To address non-stationary property in wireless networks, the author in [80] intends to use Deep RL algorithms rather than the standard RL proposed in the paper. To study the impact of mobile WSNs in the data collection problem with a mobile agent, the authors in [98] try to dynamically adjust the network structure while always ensuring a lower energy consumption. The evaluation of sensor nodes under different mobility scenarios has been also mentioned as a future work in [64, 92] to study their influences on performances.

- Resource constraints, in particular energy-saving, is a fundamental issue in developing wireless sensor systems with the goal to extend the network life-time. Thus, the employed RL and DRL techniques should minimize algorithms' complexity in terms of memory space and reduce their execution time when running them on IoT devices. Low-power and lightweight RL and DRL frameworks, such as ElegantRL [120] and TensorFlow Lite [121] have been designed to run on sensor and mobile devices with mostly equivalent features as heavy solutions, while optimizing performance and reducing binary file size. Another interesting feature to optimize and reduce the required re-

sources is the deployment model of the RL agents. Compared to centralized approaches, which rely on a single network node, decentralized approaches share the learning computation load among various wireless nodes. Using distributed approaches enables RL agents to avoid the overhead of running tasks by observing and predicting only its own environment.

7. Conclusion

This survey presented recent publications that applied both RL and DRL techniques in wireless IoT environments. First, an overview of wireless networks, MDP, RL, and DRL techniques is provided. Then, we presented a taxonomy based on IoT networking and application problems, including routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and edge caching. Additionally, we summarized for each paper the used method, the state space, the action space, and the reward function. Afterwards, we studied the proposed contributions in terms of time complexity, energy consumption, designed systems, and evaluation methods, followed by statistical analysis. Finally, we identified the remaining challenges and open issues for applying RL and DRL techniques in IoT. It is important to emphasize that these techniques are valuable to solve many issues in IoT networking and communication operations, but more work is required to cover more management operations such us monitoring, configuration and the security of these environments.

List of Abbreviations

$5G$	5th-Generation wireless systems
$A2C$	Advantage Actor-Critic
$A3C$	Asynchronous Advantage Actor-Critic
$AIoT$	Autonomous Internet of Things
ANN	Artificial Neural Network
BLE	Bluetooth Low Energy
$CIoT$	Cognitive Internet of Things
CNN	Convolutional Neural Network
CRL	Cooperative Reinforcement Learning
$CSMA/CA$	Carrier Sense Multiple Access with Collision Avoidance
DL	Deep Learning
DNN	Deep Neural Network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
DSA	Dynamic Spectrum Access
EH	Energy Harvesting
$EOSPF$	Enhanced Open Shortest Path First
GPS	Global Positioning System
$IIoT$	Industrial Internet of Things
IoT	Internet of Things
IoV	Internet of Vehicles
IP	Internet Protocol
LPS	Local Positioning System
$M2M$	Machine-to-Machine
MAC	Media Access Control
$MARL$	Multi Agent Reinforcement Learning
$MCTS$	Monte Carlo Tree Search
MDP	Markov Decision Process
MEC	Mobile Edge Computing
ML	Machine Learning
$OSPF$	Open Shortest Path First
PDR	Packet Delivery Ratio
$POMDP$	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization
QoE	Quality of Experience
QoS	Quality of Service
RA	Resource Allocation
$REINFORCE$	REward Increment = Nonnegative Factor x Offset Reinforcement x Characteristic Eligibility
RL	Reinforcement Learning
RPL	Routing Protocol for Low-Power and Lossy Networks
$RSSI$	Received Signal Strength Indicator
$SARL$	Single-Agent Reinforcement Learning
$SARSA$	State–Action–Reward–State–Action
SDN	Software Defined Network
$TDMA$	Time Division Multiple Access
$TRPO$	Trust Region Policy Optimization
$UOWSN$	Underwater Optical Wireless Sensor Network
UWB	Ultra Wide-Band
$UWSN$	Underwater Wireless Sensor Network
$VANET$	Vehicular Ad-Hoc Network
$WBAN$	Wireless Body Area Networks
$WPCD$	Wireless Portable Charging Device
WSN	Wireless Sensor Network

References

[1] J. Ding, M. Nemati, C. Ranaweera, J. Choi, Iot connectivity technologies and applications: A survey, IEEE Access (2020).

[2] R. Porkodi, V. Bhuvaneswari, The internet of things (iot) applications and communication enabling technology standards: An overview, in: 2014 International conference on intelligent computing applications, IEEE, 2014, pp. 324–329.

[3] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, K.-S. Kwak, The internet of things for health care: a comprehensive survey, IEEE access 3 (2015) 678–708.

[4] M. R. Jabbarpour, A. Nabaei, H. Zarrabi, Intelligent guardrails: an iot application for vehicle traffic congestion reduction in smart city, in: 2016 ieee international conference on internet of things (ithings) and ieee green computing and communications (greencom) and ieee cyber, physical and social computing (cpscom) and ieee smart data (smartdata), IEEE, 2016, pp. 7–13.

[5] M. Abbasi, M. H. Yaghmaee, F. Rahnama, Internet of things in agriculture: A survey, in: 2019 3rd International Conference on Internet of Things and Applications (IoT), IEEE, 2019, pp. 1–12.

[6] U. Z. A. Hamid, H. Zamzuri, D. K. Limbu, Internet of vehicle (iov) applications in expediting the implementation of smart highway of autonomous vehicle: A survey, in: Performability in Internet of Things, Springer, 2019, pp. 137–157.

[7] T. M. Mitchell, et al., Machine learning. 1997, 432, McGraw-Hill, 1997.

[8] A. Thessen, Adoption of machine learning techniques in ecology and earth science, One Ecosystem 1 (2016) e8621.

[9] M. Vakili, M. Ghamsari, M. Rezaei, Performance analysis and comparison of machine and deep learning algorithms for iot data classification, arXiv preprint arXiv:2001.09636 (2020).

[10] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2017.
[11] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.
[12] J. Schmidhuber, Deep learning in neural networks: An overview, Neural networks 61 (2015) 85–117.
[13] B. Yegnanarayana, Artificial neural networks, PHI Learning Pvt. Ltd., 2009.
[14] H. A. Al-Rawi, M. A. Ng, K.-L. A. Yau, Application of reinforcement learning to routing in distributed wireless networks: a review, Artificial Intelligence Review 43 (2015) 381–416.
[15] I. Althamary, C.-W. Huang, P. Lin, A survey on multi-agent reinforcement learning methods for vehicular networks, in: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2019, pp. 1154–1159.
[16] Y. Wang, Z. Ye, P. Wan, J. Zhao, A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks, Artificial Intelligence Review 51 (2019) 493–506.
[17] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, D. I. Kim, Applications of deep reinforcement learning in communications and networking: A survey, IEEE Communications Surveys & Tutorials 21 (2019) 3133–3174.
[18] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, X. Shen, Deep reinforcement learning for autonomous internet of things: Model, applications and challenges, IEEE Communications Surveys & Tutorials (2020).
[19] T. T. Nguyen, N. D. Nguyen, S. Nahavandi, Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications, IEEE transactions on cybernetics (2020).
[20] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, J. Qin, A survey on application of machine learning for internet of things, International Journal of Machine Learning and Cybernetics 9 (2018) 1399–1417.
[21] K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, V. H. C. de Albuquerque, Internet of things: A survey on machine learning-based intrusion detection approaches, Computer Networks 151 (2019) 147–157.
[22] D. P. Kumar, T. Amgoth, C. S. R. Annavarapu, Machine learning algorithms for wireless sensor networks: A survey, Information Fusion 49 (2019) 1–25.
[23] IEEE standard for local and metropolitan area networks - part 15.6: Wireless body area networks, IEEE Std 802.15.6-2012 (2012).
[24] M. Patel, J. Wang, Applications, challenges, and prospective in emerging body area networking technologies, IEEE Wireless communications 17 (2010) 80–88.
[25] A. Gkikopouli, G. Nikolakopoulos, S. Manesis, A survey on underwater wireless sensor networks and applications, in: 2012 20th Mediterranean conference on control & automation (MED), IEEE, 2012, pp. 1147–1154.
[26] H. Kaushal, G. Kaddoum, Underwater optical wireless communication, IEEE access 4 (2016) 1518–1547.
[27] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular ad hoc network, Journal of network and computer applications 37 (2014) 380–392.
[28] F. Yang, S. Wang, J. Li, Z. Liu, Q. Sun, An overview of internet of vehicles, China communications 11 (2014) 1–15.
[29] D. Jiang, L. Delgrossi, Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments, in: VTC Spring 2008-IEEE Vehicular Technology Conference, IEEE, 2008, pp. 2036–2040.
[30] A. Gilchrist, Industry 4.0: the industrial internet of things, Springer, 2016.
[31] M. T. Spaan, Partially observable markov decision processes, in: Reinforcement Learning, Springer, 2012, pp. 387–414.
[32] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, et al., Imagination-augmented agents for deep reinforcement learning, in: Advances in neural information processing systems, 2017, pp. 5690–5701.
[33] A. Nagabandi, G. Kahn, R. S. Fearing, S. Levine, Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 7559–7566.
[34] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science 362 (2018) 1140–1144.
[35] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of Monte Carlo tree search methods, IEEE Transactions on Computational Intelligence and AI in games 4 (2012) 1–43.
[36] M. H. Kalos, P. A. Whitlock, Monte carlo methods, John Wiley & Sons, 2009.
[37] G. A. Rummery, M. Niranjan, On-line Q-learning using connectionist systems, volume 37, University of Cambridge, Department of Engineering Cambridge, UK, 1994.
[38] C. J. Watkins, P. Dayan, Q-learning, Machine learning 8 (1992) 279–292.
[39] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: Advances in neural information processing systems, 2000, pp. 1057–1063.
[40] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (1992) 229–256.
[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International conference on machine learning, 2015, pp. 1889–1897.
[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).
[43] V. R. Konda, J. N. Tsitsiklis, Actor-critic algorithms, in: Advances in neural information processing systems, 2000, pp. 1008–1014.
[44] I. Grondman, L. Busoniu, G. A. Lopes, R. Babuska, A survey of actor-critic reinforcement learning: Standard and natural policy gradients, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (2012) 1291–1307.
[45] K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Processing Magazine 34 (2017) 26–38.
[46] L. Deng, D. Yu, et al., Deep learning: methods and applications, Foundations and Trends® in Signal Processing 7 (2014) 197–387.
[47] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.
[48] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).
[49] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.
[50] R. Caruana, Multitask learning, Machine learning 28 (1997) 41–75.
[51] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, arXiv preprint arXiv:1703.03400 (2017).
[52] L. Ericsson, More than 50 billion connected devices, White Paper 14 (2011) 124.
[53] J. Chase, The evolution of the internet of things, Texas Instruments 1 (2013) 1–7.
[54] E. Ancillotti, C. Vallati, R. Bruno, E. Mingozzi, A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management, Computer Communications 112 (2017) 1–13.
[55] J. Yang, S. He, Y. Xu, L. Chen, J. Ren, A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks, Sensors 19 (2019) 970.
[56] X.-h. Liu, D.-g. Zhang, T. Zhang, Y.-y. Cui, Novel approach of the best path selection based on prior knowledge reinforcement learning, in: 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), IEEE, 2019, pp. 148–154.
[57] X.-h. Liu, D.-g. Zhang, T. Zhang, Y.-y. Cui, New method of the best path selection with length priority based on reinforcement learning strategy, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2019, pp. 1–6.
[58] G. Liu, X. Wang, X. Li, J. Hao, Z. Feng, ESRQ: An efficient secure routing method in wireless sensor networks based on Q-learning, in: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE,

2018, pp. 149–155.
[59] L. Zhao, J. Wang, J. Liu, N. Kato, Routing for crowd management in smart cities: A deep reinforcement learning perspective, IEEE Communications Magazine 57 (2019) 88–93.
[60] W. Guo, C. Yan, T. Lu, Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing, International Journal of Distributed Sensor Networks 15 (2019) 1550147719833541.
[61] Y. Akbari, S. Tabatabaei, A new method to find a high reliable route in IoT by using reinforcement learning and fuzzy logic, Wireless Personal Communications (2020) 1–17.
[62] N. Aslam, K. Xia, M. U. Hadi, Optimal wireless charging inclusive of intellectual routing based on SARSA learning in renewable wireless sensor networks, IEEE Sensors Journal 19 (2019) 8340–8351.
[63] A. Mateen, M. Awais, N. Javaid, F. Ishmanov, M. K. Afzal, S. Kazmi, Geographic and opportunistic recovery with depth and power transmission adjustment for energy-efficiency and void hole alleviation in UWSNs, Sensors 19 (2019) 709.
[64] H. Chang, J. Feng, C. Duan, Reinforcement learning-based data forwarding in underwater wireless sensor networks with passive mobility, Sensors 19 (2019) 256.
[65] S. Wang, Y. Shin, Efficient routing protocol based on reinforcement learning for magnetic induction underwater sensor networks, IEEE Access 7 (2019) 82027–82037.
[66] X. Li, X. Hu, W. Li, H. Hu, A multi-agent reinforcement learning routing protocol for underwater optical sensor networks, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), IEEE, 2019, pp. 1–7.
[67] M. Kwon, J. Lee, H. Park, Intelligent IoT connectivity: deep reinforcement learning approach, IEEE Sensors Journal (2019).
[68] H. V. Hasselt, Double Q-learning, in: Advances in neural information processing systems, 2010, pp. 2613–2621.
[69] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in: Thirtieth AAAI conference on artificial intelligence, 2016.
[70] Z. Wei, Y. Zhang, X. Xu, L. Shi, L. Feng, A task scheduling algorithm based on q-learning and shared value function for WSNs, Computer Networks 126 (2017) 141–149.
[71] M. I. Khan, K. Xia, A. Ali, N. Aslam, Energy-aware task scheduling by a true online reinforcement learning in wireless sensor networks., IJSNet 25 (2017) 244–258.
[72] Z. Wei, F. Liu, Y. Zhang, J. Xu, J. Ji, Z. Lyu, A Q-learning algorithm for task scheduling based on improved SVM in wireless sensor networks, Computer Networks 161 (2019) 138–149.
[73] S. Kosunalp, Y. Chu, P. D. Mitchell, D. Grace, T. Clarke, Use of Q-learning approaches for practical medium access control in wireless sensor networks, Engineering Applications of Artificial Intelligence 55 (2016) 146–154.
[74] Y. Chu, P. D. Mitchell, D. Grace, ALOHA and Q-learning based medium access control for wireless sensor networks, in: 2012 International Symposium on Wireless Communication Systems (ISWCS), IEEE, 2012, pp. 511–515.
[75] T. Yang, J. Li, H. Feng, N. Cheng, W. Guan, A novel transmission scheduling based on deep reinforcement learning in software-defined maritime communication networks, IEEE Transactions on Cognitive Communications and Networking 5 (2019) 1155–1166.
[76] H. Yang, X. Xie, An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive internet of things systems, IEEE Systems Journal (2019).
[77] Y. Lu, T. Zhang, E. He, I.-S. Comşa, Self-learning-based data aggregation scheduling policy in wireless sensor networks, Journal of Sensors 2018 (2018).
[78] R. F. Atallah, C. M. Assi, M. J. Khabbaz, Scheduling the operation of a connected vehicular network using deep reinforcement learning, IEEE Transactions on Intelligent Transportation Systems 20 (2018) 1669–1682.
[79] Y.-H. Xu, J.-W. Xie, Y.-G. Zhang, M. Hua, W. Zhou, Reinforcement learning (RL)-based energy efficient resource allocation for energy harvesting-powered wireless body area network, Sensors 20 (2020) 44.
[80] A. Gomes, D. F. Macedo, L. F. Vieira, Automatic MAC protocol selection in wireless networks based on reinforcement learning, Computer Communications 149 (2020) 312–323.
[81] C. Wang, X. Gaimu, C. Li, H. Zou, W. Wang, Smart mobile crowdsensing with urban vehicles: A deep reinforcement learning perspective, IEEE Access 7 (2019) 37334–37341.
[82] Y. Liu, H. Yu, S. Xie, Y. Zhang, Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks, IEEE Transactions on Vehicular Technology 68 (2019) 11158–11168.
[83] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, J. Hu, iRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks, IEEE Internet of Things Journal 6 (2019) 7011–7024.
[84] S. S. Oyewobi, G. P. Hancke, A. M. Abu-Mahfouz, A. J. Onumanyi, An effective spectrum handoff based on reinforcement learning for target channel selection in the industrial internet of things, Sensors 19 (2019) 1395.
[85] C. Fan, S. Bao, Y. Tao, B. Li, C. Zhao, Fuzzy reinforcement learning for robust spectrum access in dynamic shared networks, IEEE Access 7 (2019) 125827–125839.
[86] D. Pacheco-Paramo, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, Deep reinforcement learning mechanism for dynamic access control in wireless networks handling mMTC, Ad Hoc Networks 94 (2019) 101939.
[87] S. Wang, H. Liu, P. H. Gomes, B. Krishnamachari, Deep reinforcement learning for dynamic multichannel access in wireless networks, IEEE Transactions on Cognitive Communications and Networking 4 (2018) 257–265.
[88] M. Chincoli, A. Liotta, Self-learning power control in wireless sensor networks, Sensors 18 (2018) 375.
[89] C. Savaglio, P. Pace, G. Aloi, A. Liotta, G. Fortino, Lightweight reinforcement learning for energy efficient communications in wireless sensor networks, IEEE Access 7 (2019) 29355–29364.
[90] S. Soni, M. Shrivastava, Novel learning algorithms for efficient mobile sink data collection using reinforcement learning in wireless sensor network, Wireless Communications and Mobile Computing 2018 (2018).
[91] F. A. Aoudia, M. Gautier, O. Berder, Learning to survive: Achieving energy neutrality in wireless sensor networks using reinforcement learning, in: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1–6.
[92] Y. Wu, K. Yang, Cooperative reinforcement learning based throughput optimization in energy harvesting wireless sensor networks, in: 2018 27th Wireless and Optical Communication Conference (WOCC), IEEE, 2018, pp. 1–6.
[93] A. Murad, F. A. Kraemer, K. Bach, G. Taylor, Autonomous management of energy-harvesting IoT nodes using deep reinforcement learning, in: 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2019, pp. 43–51.
[94] M. K. Sharma, A. Zappone, M. Debbah, M. Assaad, Multi-agent deep reinforcement learning based power control for large energy harvesting networks, in: Proc. 17th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw.(WiOpt), 2019, pp. 1–7.
[95] X. Wang, Q. Zhou, C. Qu, G. Chen, J. Xia, Location updating scheme of sink node based on topology balance and reinforcement learning in WSN, IEEE Access 7 (2019) 100066–100080.
[96] M. Mohammadi, A. Al-Fuqaha, M. Guizani, J.-S. Oh, Semisupervised deep reinforcement learning in support of IoT and smart city services, IEEE Internet of Things Journal 5 (2017) 624–635.
[97] C. H. Liu, Q. Lin, S. Wen, Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning, IEEE Transactions on Industrial Informatics 15 (2018) 3516–3526.
[98] J. Lu, L. Feng, J. Yang, M. M. Hassan, A. Alelaiwi, I. Humar, Artificial agent: The fusion of artificial intelligence and a mobile agent for energy-efficient traffic control in wireless sensor networks, Future Generation Computer Systems 95 (2019) 45–51.
[99] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, S. Jin, Caching transient data for internet of things: A deep reinforcement learning approach, IEEE Internet of Things Journal 6 (2018) 2074–2083.
[100] Y. Wei, F. R. Yu, M. Song, Z. Han, Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning, IEEE Internet of Things Journal 6 (2018) 2061–2073.
[101] W. Liang, W. Huang, J. Long, K. Zhang, K.-C. Li, D. Zhang, Deep reinforcement learning for resource protection and real-time detection in IoT environment, IEEE Internet of Things Journal (2020).
[102] J. Banks, Introduction to simulation, in: Proceedings of the 31st confer-

ence on Winter simulation: Simulation—a bridge to the future-Volume 1, 1999, pp. 7–13.
[103] Mathworks - solutions - MATLAB & simulink, [Accessed on July 2020]. URL: https://fr.mathworks.com/solutions.html.
[104] Products and services - MATLAB & simulink, [Accessed on July 2020]. URL: https://www.mathworks.com/products.html.
[105] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with cooja, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, IEEE, 2006, pp. 641–648.
[106] OMNeT++ discrete event simulator, [Accessed on July 2020]. URL: https://omnetpp.org/.
[107] The network simulator - NS-2, [Accessed on July 2020]. URL: https://www.isi.edu/nsnam/ns/.
[108] NS-3 — a discrete-event network simulator for internet systems, [Accessed on July 2020]. URL: https://www.nsnam.org/.
[109] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), 2016, pp. 265–283.
[110] Gym: A toolkit for developing and comparing reinforcement learning algorithms, [Accessed on July 2020]. URL: https://gym.openai.com/.
[111] N. H. Mak, W. K. Seah, How long is the lifetime of a wireless sensor network ?, in: 2009 International Conference on Advanced Information Networking and Applications, IEEE, 2009, pp. 763–770.
[112] A. P. Renold, S. Chandrakala, MRL-SCSO: Multi-agent reinforcement learning-based self-configuration and self-optimization protocol for unattended wireless sensor networks, Wireless Personal Communications 96 (2017) 5061–5079.
[113] R. S. Sutton, Learning to predict by the methods of temporal differences, Machine learning 3 (1988) 9–44.
[114] A. Y. Ng, S. J. Russell, et al., Algorithms for inverse reinforcement learning., in: Icml, volume 1, 2000, p. 2.
[115] P. Abbeel, A. Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: Proceedings of the twenty-first international conference on Machine learning, 2004, p. 1.
[116] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, A. Dragan, Inverse reward design, in: Advances in neural information processing systems, 2017, pp. 6765–6774.
[117] S. Levine, A. Kumar, G. Tucker, J. Fu, Offline reinforcement learning: Tutorial, review, and perspectives on open problems, arXiv preprint arXiv:2005.01643 (2020).
[118] Y. Wu, E. Mansimov, S. Liao, A. Radford, J. Schulman, Openai baselines: ACKTR and A2C, 2017. URL: https://openai.com/blog/baselines-acktr-a2c/.
[119] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: International conference on machine learning, 2016, pp. 1928–1937.
[120] Elegantrl: Lightweight, efficient and stable deep reinforcement learning implementation using pytorch, [Accessed on May 2021]. URL: https://github.com/AI4Finance-LLC/ElegantRL.
[121] Tensorflow lite — ml for mobile and edge devices, [Accessed on May 2021]. URL: https://www.tensorflow.org/lite.