Hindawi
Complexity
Volume 2021, Article ID 7179374, 31 pages
https://doi.org/10.1155/2021/7179374

Review Article

The Applicability of Reinforcement Learning Methods in the Development of Industry 4.0 Applications

Tamás Kegyes,[1] **Zoltán Süle**,[2] **and János Abonyi**[1]

[1]*MTA-PE "Lendület" Complex Systems Monitoring Research Group, University of Pannonia, Veszprém, Hungary*
[2]*University of Pannonia, Veszprém, Hungary*

Correspondence should be addressed to János Abonyi; janos@abonyilab.com

Received 9 September 2021; Accepted 25 October 2021; Published 30 November 2021

Academic Editor: Murari Andrea

Reinforcement learning (RL) methods can successfully solve complex optimization problems. Our article gives a systematic overview of major types of RL methods, their applications at the field of Industry 4.0 solutions, and it provides methodological guidelines to determine the right approach that can be fitted better to the different problems, and moreover, it can be a point of reference for R&D projects and further researches.

1. Introduction

Reinforcement learning (RL) has a significant chance to revolutionize the artificial intelligence (AI) applications by serving a novel approach of machine learning (ML) developments that lets the user to handle large-scale problems efficiently. These techniques together with widespread Internet of things tools have opened up new possibilities for optimizing complex systems, including domains of logistics, project planning, scheduling, and further industry-related domains. Extracting this potential can result in a fundamental progress of Industry 4.0 transformation [1]. During this digital transformation, the vertical and horizontal integration will be strengthened, the flexibility should be raised, and the human control and supervision need to be focused [2, 3]. Furthermore, the data produced by the integrated tools are increasing exponentially that requires a higher level of autonomous processes and decisions. Reinforcement learning can serve as a valuable tool in the development of self-optimising and organising Industry 4.0 solutions. The main challenge of developing these applications is that there are several methods and techniques and a wide range of parameters that need to be defined. As the definition of these parameters requires detailed knowledge of the nature of the RL algorithms, the main goal of this paper is to provide a comprehensive overview of RL methods from the viewpoint of Industry 4.0 and smart manufacturing.

On the basis of our best knowledge, there exists no similar overview article of reinforcement learning methods in Industry 4.0 applications. Next to the fundamental book [4], there are several overviews of reinforcement learning methods from theoretical point of view. A detailed semantic overview of Industry 4.0 frameworks [5] and a categorization of Industry 4.0 research fields are also described. An overview of key elements of Industry 4.0 researches and several application scenarios [6] highlighted the wide scope of smart manufacturing. Although many authors found that there is a lack of extensive review of Industry 4.0 revolution from different aspects, according to their persistent work nowadays, several articles are available in this topic [7]. A survey on the applications of optimal control to scheduling in production, supply chain, and Industry 4.0 systems [8] focused on maximum principle-based studies. Most of the surveys and review articles of Industry 4.0 declare the importance of optimization, but mostly only general approaches are discussed, and there are no detailed guidelines extracted. A comprehensive survey at field of Industry 4.0 and optimization [9] discussed the recent developments in data fusion and machine learning for industrial prognosis, placing an emphasis on the identification of research trends, niches of opportunity, and unexplored challenges. Even if it

considered several ML methods and algorithms, RL was mentioned only shortly without extracting its key fundamentals.

The above collected facts strengthened our motivation of preparing a detailed overview of RL applications and methods used in the field of Industry 4.0. Our main goals with this are:

Presenting a hands-on reference for researchers who are interested in RL applications

Giving compact descriptions of applicable RL methods

Serving a guideline to support them easily identify the best fitting subset of RL methods to their problems and hence letting them focus on the relevant part of the literature

Our systematic review is based on an examination of the literature available from Scopus by following the PRISMA-P (Preferred Reporting Items for Systematic Reviews and Meta-Analysis Protocols). The PRISMA-P workflow contains a 17-item checklist that supports to facilitate the preparation and reporting of a robust protocol in a standardized way for systematic reviews. The literature source list was queried in February 2021 with the following keywords: TITLE-ABS-KEY ("reinforcement learning" AND ("smart factory" OR "IOT" OR "smart manufacturing" OR "industry 4.0" OR "CPS")).

Both author keywords and index keywords were involved into the analysis. The keyword processing started with an extensive data cleansing process by:

Building up a standardized keyword unit (SKU) list and splitting complex keywords into SKUs

Assigning SKUs to one of the following keyword classification types:

(i) Principle captured
(ii) Industrial field of application
(iii) Application field of solution
(iv) Mathematical approach of application methodology

Identifying major classification groups by classification types

781 articles were involved into the analysis. Out of 14,035 original author and index keywords, 2,579 duplications were filtered out. The remaining 11,456 keywords were sliced into 45,824 SKUs. Finally, 12,017 keywords were assigned to classification types that provide the major tendencies and relations of industrial applications of reinforcement learning methods. Figure 1 shows the change of the assessed literature size over the PRISMA steps.

Our article stands for the following major parts:

First, in Section 2, we will give a short general introduction of reinforcement learning framework and summarize some major mathematical properties behind RL techniques. Furthermore, we will present a classification of RL methods that lets the reader to have a map for the further discussions.

As a next step in Sections 3.1–3.3, we will present the key findings of systematic review and a hands-on reference for further researches.

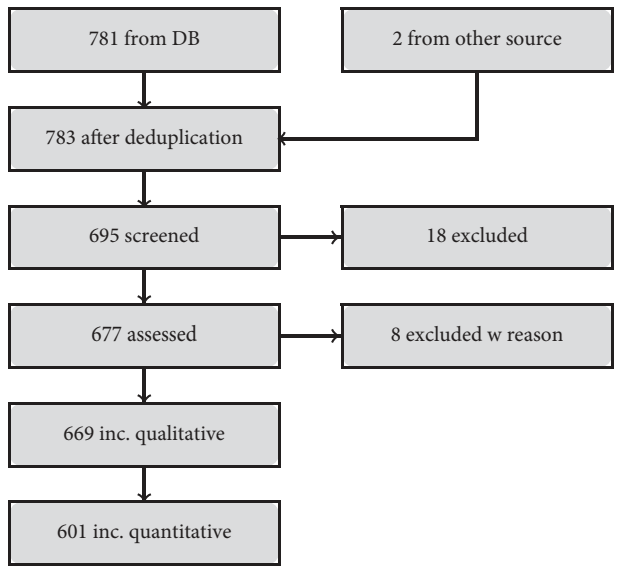

Figure 1: PRISMA processing flow.

Then, in Section 3.4 and in Section 3.5, we will discuss the conclusions and give a detailed guideline to help the reader to choose to most adequate RL method for the different problems.

Finally, in Appendices A–H, we will provide a compact overview of 18 different RL methods.

2. Theoretical Background of Reinforcement Learning

In this section, we will summarize the fundamental concept of reinforcement learning, then we will present a general classification of RL methods.

There are three main paradigms in machine learning: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, a functional relationship of a regression model of a classifier is learnt based on data that represent the input and output of the model. In unsupervised learning, the hidden structure of the data is explored, usually by clustering [9].

Reinforcement learning (RL) also refers to learning problems. As Figure 2 represents the process, an agent takes observations of the environment; then on the basis of that, it executes an action (A_t). As a result of the action in the environment, the agent will get a reward (R_t) and it can take a new observation (O_t) from the environment and the cycle is repeated. The problem is to let agent learning so as to maximize the total reward. Reinforcement learning concept was introduced in ([4], Section 3.1). While in supervised and unsupervised learning, the model fitting requires a complete set of observations; in reinforcement learning, the learning process is sequential.Reinforcement learning is based on the reward hypothesis which states that all goals can be described by the maximisation of expected cumulative rewards. Formally, the history is the sequence of observations, actions, and rewards: $H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$.

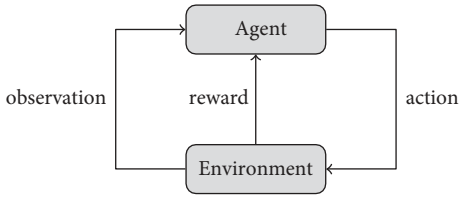

Figure 2: Reinforcement learning framework.

A state contains all the information to determine what happens next. Formally, state is a function of the history: $S_t = f(H_t)$. Let G_t denote the total discounted reward from time-step t: $G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$.

The state-value function $v(s)$ gives the expected total discounted return if starting from state s: $v(s) = \mathbb{E}[G_t \mid S_t t = ns]$. Policy covers the agent's behaviour in all possible cases, so it is essentially a map from states to actions. There are two major categories in it: (1) deterministic policy: $a = \pi(s)$, (2) stochastic policy: $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$. The action-value function $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π: $q_\pi(s; a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$.

Practically, state-value function is a prediction of expected present values (PV) of future rewards that allows evaluating the goodness of states, so it is a map from states to scalars: $v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s]$. The optimal state-value function $v^*(s)$ is the maximum state-value function overall policies: $v^*(s) = \max_\pi v_\pi(s)$. It is easy to find that in case if an optimal state-value function is known that an optimal action-value function and an optimal policy can be derived.

Reinforcement learning concept is based on stochastic processes and on Markov chains. Markov property is fundamental of mathematical basis of reinforcement learning methods. A state is Markov if and only if the $\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \ldots, S_t]$ condition holds. By definition, a Markov decision process (MDP) is a tuple of $\langle \mathcal{S}; \mathcal{A}; \mathrm{P}; \mathcal{R}; \gamma \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, P is a state transition probability matrix, $\mathrm{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$, $\mathcal{R}$ is a reward function, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$, γ is a discount factor, $\gamma \in [0; 1]$, and t time-steps are discrete. The Bellman equation practically states that state-value function of an MDP can be decomposed into two parts: immediate reward and discounted value of successors states: $v(s = S_t) = R_{t+1} + \gamma v(S_{t+1})$.

Environments can be distinguished by its observability. Let us denote S^a_t as the agent's state at time-step t and S^e_t as the environment's state. Environment can be (1) fully observable if the agent directly observes all states of environment $(O_t = S^a_t = S^e_t)$, or partially observable if the agent has indirect observations $(S^a_t = (\mathbb{P}[S^e_t = s_1], \ldots, \mathbb{P}[S^e_t = s_n]))$.

Figure 3 summarizes a classification of reinforcement learning methods in tree structure. Further details of the different RL methods are described in Appendix.

3. Overview of the Industry 4.0 Relevant Applications

In this section, we will present the hands-on references in tabular format based on the results of our data cleansing process and some major results of systematic literature analysis that will highlight some general trends which is able to lead the reader to a successfully applicable RL methods by preventing the usage of inappropriate trials and hence shortening development periods. In the final part of the section, we will present a hands-on guideline to summarize the key conclusions.

3.1. Classification of Applications by Principle Captured. The main goal of this section is to give an overview what are the principal captured problem types that reinforcement learning was applied for and describe the major tools that gave an impressive performance for each and every problem category and finally to highlight some typical issues that needed to be taken care of during the implementation.

By performing SKU analysis, we identified the most relevant keywords that are assigned to a principle captured. In Table 1, the associated publications are listed by principle captured categories.

Furthermore, Figure 4 shows the principle captured classes by reinforcement learning methods. Although the related frequency table does not meet all the required criteria, in Table 2, a χ^2-test, calculation is presented, yet by principle captured classes, it makes the identification of some significant deviations from the overall distribution of RL methods possible.

In the class of prediction, forecasting, and estimation, planning value function approximation methods and Markov decision processes are over-represented. This lets us to conclude that the complex methods in the focus are less, which is fully in line with the goal to understand better the behaviour of the environment without strong optimization aims.

In the class of detection, recognition, prevention, avoidance, and protection, the policy gradient methods are over-represented, while MDPs are under-represented. This shows us that researchers are interested more in complex models with a higher predictive performance than in basic solutions.

In the classes of evaluation, assessment and allocation, assignment, and resource management, the multiagent methods are more in focus, which tells us that this field is on the way to distribute the tasks to lower level tools instead of centralized data processes. But while in the first class, the distribution of further RL methods follows the overall distribution, and in the second class, the policy gradient methods are over-represented which comes from the fact that allocation-related problems prefer to create an optimal policy.

In the classes of classification, clustering and decision making and scheduling, queuing, and planning, the situation is opposite: multiagent methods are under-represented, which means that researches of these kinds of operations are still focusing to a centralized solution.

In the class of control, the temporal-difference methods and Markov decision process contractions and multi-agent methods are over-represented, while complex approaches, like policy gradient methods, are under-represented.

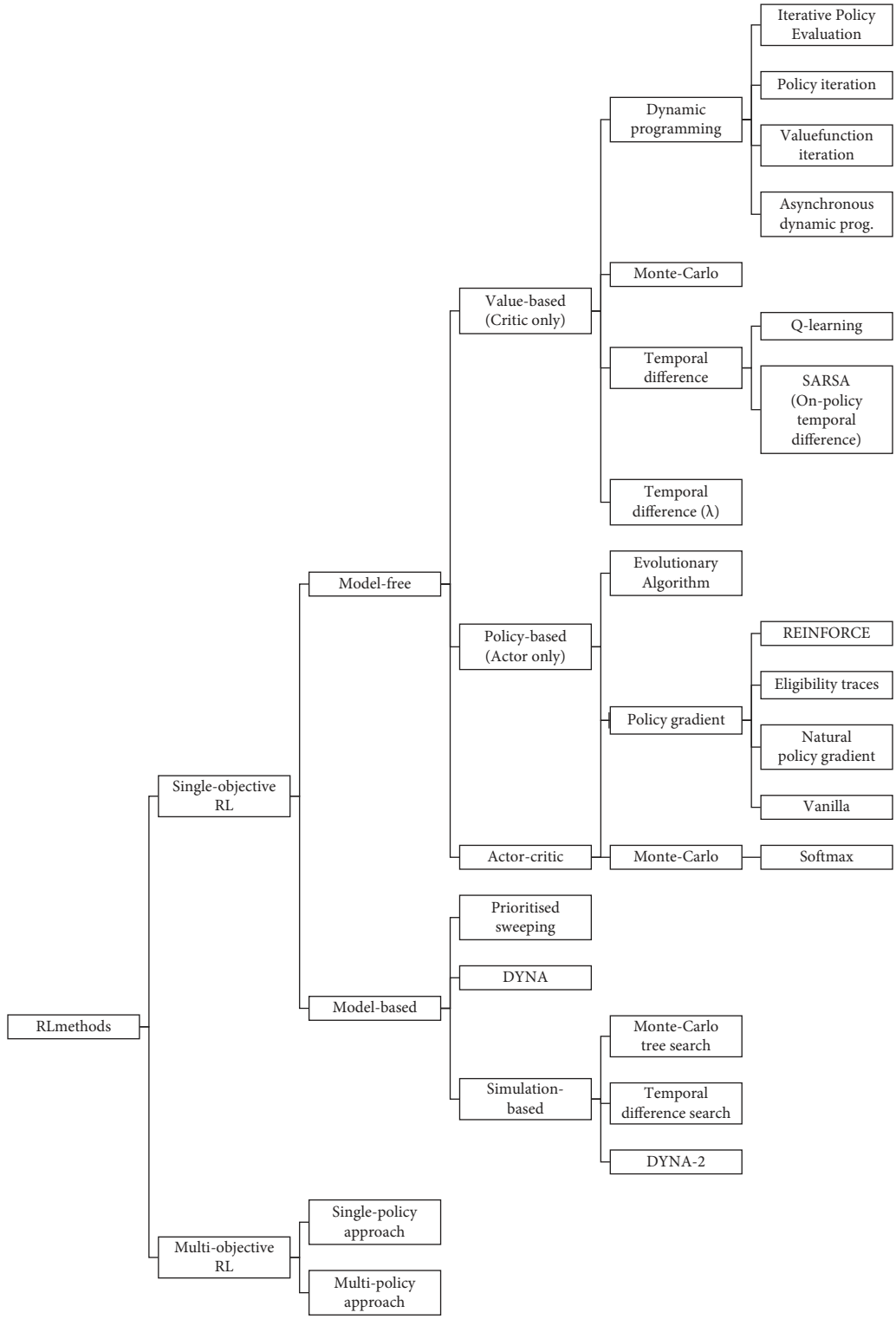

Figure 3: Classification tree of reinforcement learning methods.

Discussions of specific parts of RL solution design problems occur in smaller number of cases, but these kinds of publication demonstrate the fact that constructing an appropriate RL application is not always trivial. We can highlight state space design [12, 25, 33, 107, 144, 179, 193, 208, 217, 220, 222, 224, 227, 266, 267] and action space design [109, 220, 246, 268], reward construction [14, 76, 110, 199, 220, 226, 246, 269–273], and exploration strategy planning [86, 274] which can be determinants from the whole application point of view.

Table 1: Publication reference by principle captured.

Principle captured	Referred publications
Prediction, forecasting, estimation, planning	[10–39]
Detection, recognition, prevention, avoidance, protection	[10, 29, 39–71]
Evaluation, assessment	[18, 32, 54, 66, 72–81]
Classification, clustering	[35, 42, 66, 69, 69, 81–95]
Decision making	[11, 13, 17, 20, 21, 24, 38, 43, 61, 62, 66, 69, 82, 89, 93], [96–131]
Allocation, assignment, resource management	[20, 22, 31, 32, 39, 45, 60, 65, 67, 70, 75, 78, 83, 87, 91, 96, 97, 99, 100], [103, 104, 113, 119, 121, 121, 125, 127, 130], [130, 131, 131–154], [154–156, 156–178], [179–196], [196–202]
Scheduling, queuing, planning	[12, 19, 21, 24, 32, 72, 87, 88, 91, 93, 96, 99, 110, 113, 122, 125, 131, 150, 151, 160, 184, 188, 203–222]
Control	[12, 14, 15, 18, 23, 27, 31, 36, 37, 40, 56, 69, 70, 91, 93, 99, 101], [105–107, 111, 112, 122, 123, 129, 130], [149, 150, 167–169, 171, 180], [188, 189, 199, 200, 205, 217], [223–245], [245–265]

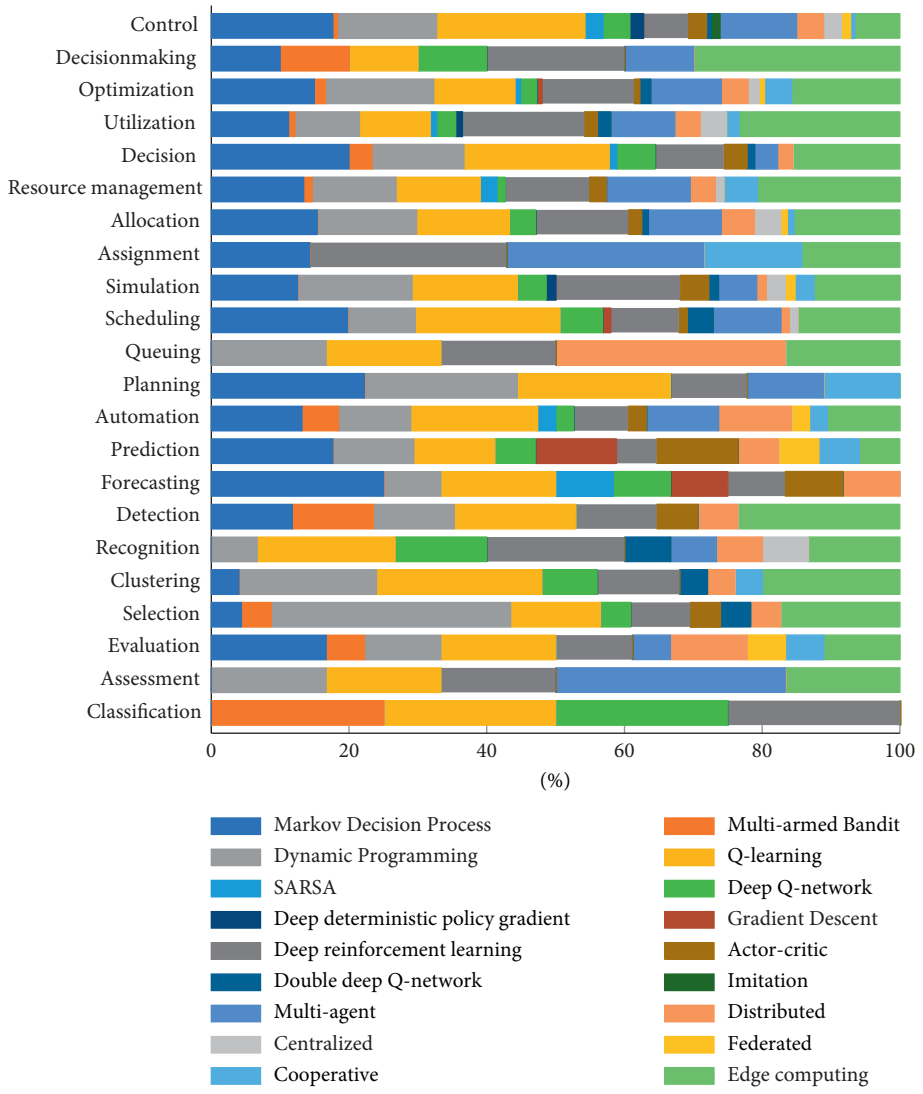

Figure 4: Distribution of RL methods by principal captured classes.

Table 2: χ^2-test table of principle captured classes by RL method types.

Principle captured	Markov decision process	Multiarmed bandit	Dynamic	Temporal difference	Value function approximation	Policy gradient	Multiagent	Edge computing
Prediction, forecasting, estimation, planning	1.96	−0.58	0.12	−0.32	3.08	0.31	−0.39	−4.17
Detection, recognition, prevention, avoidance, protection	−3.09	1.51	−1.11	−0.17	0.9988, 0.9782, 0.97850.39	0.9935, 0.8769, 0.87852.2	−1.38	1.65
Evaluation, assessment	−0.82	0.63	−0.08	−0.62	−1.21	−0.6	2.96	−0.27
Classification, clustering	−3.61	0.56	1.28	1.41	1.54	0.65	−2.88	1.05
Decision making	3.1	2.47	−0.84	1.73	0.95	0.02	−10.82	3.39
Allocation, assignment, resource management	−2.69	−1.95	0.21	−11.18	−4.74	2.08	10.54	7.74
Scheduling, queuing, planning	2.17	−1.33	−2.17	1.24	1.61	−0.04	−2.63	1.16
Control	2.67	−1.34	2.35	7.52	−1.72	−4.93	4.27	−10.82

3.2. Classification of Publications by Industrial Field of Application. Similarly, as we have shown in Section 3.1, by performing SKU analysis, we also identified the most relevant keywords that are assigned to industrial fields. In Table 3, the associated publications are listed by industrial field categories.

Similarly, as we presented categories of principal captured, we also prepared Figure 5 that shows the industrial field classes by reinforcement learning methods. Although the related frequency table does not meet all the required criteria, in Table 4, a χ^2-test, calculation is presented, yet by industrial field classes, it makes the identification of some significant deviations from the overall distribution of RL methods possible.

In the class of energy, solar, power, electric, the applications of Q-learning methods are over-represented, while more basic methods and policy gradient methods are under-represented

In the class of telecommunication, communication, networking, internet, 5G, Wi-Fi, and mobile, the policy gradient methods are over-represented and there is a strong focus on the applications of edge computing

In the class of wireless, radio, antenna, and signal, the applications of Markov decision processes are highlighted

Similarly, in the class of vehicle, unmanned aerial vehicle, drone, and aircraft, the applications of Markov decision process are over-represented together with policy gradient methods, while the multiagent solutions are less discussed

In the classes of cyber-physical system, robot and manufacturing, and factory, the basic dynamic methods and Q-learning approaches are more popular

Finally, in the class of city and building, the multiagent methods are over-represented

3.3. Classification of Publications by Mathematical Approach of Application Methodology. Similarly, as we have shown in the previous sections, we also performed the SKU analysis for the third major dimension of keywords which is the methodological approach of the solution. The most relevant keywords were identified, and then in Table 5. the associated publications are listed by methodological approach categories.

Although it is not feasible to summarize all the different methodological approaches in details, we would like to highlight some specialities of selected cases to demonstrate how widely RL approaches are used and motivate researchers to find a solution for their problems from a new perspective.

As we described in Section 2, reinforcement learning methods are based on Markov property and hence it is fundamental to model the problems as Markov decision processes (MDPs), which is far not trivial in several cases. By formulating an MDP, we need to take care about state space design, especially guaranteeing that a state representation contains all the relevant information to evaluate a situation, or with other words anytime, when the system is in the same particular action, the environment will take its response by the same characteristic for a particular action [96, 104, 191, 203, 313, 346].

Actor-critic methods are model-free learning methods that learn both the optimal policy for taking an action and the value function for most accurate evaluating of the current state. Most of the publications discuss mainly distributed autonomous IoT device networks. In these cases, the focus is shifted towards the learning and knowledge transfer solutions:

Stochastic model of cloud-based IoT for fog computing computation offload and radio resource allocation [97].

Centralized joint resource allocation solution for handling shortage of frequency resources of cellular

Table 3: Publication reference by industrial field of application.

Industrial field	Referred publications
Energy, solar, power, electric	[14, 17, 18, 23, 31, 32, 37, 45, 51, 53, 70, 72, 75, 75, 80, 81, 83, 85, 88, 100], [101, 107, 108, 118, 123, 133, 134, 136, 144, 145, 154, 159, 161, 165, 167, 168, 171], [174–177, 179, 183, 188, 192, 198, 200], [204, 208, 210, 213, 216, 221, 222, 225–227], [230–232, 234, 235, 240, 243, 246, 248, 252, 255, 257, 259], [264–266], [275–310]
Telecommunication, communication, networking, internet, 5G, Wi-Fi, mobile	[12, 17, 22, 28, 33, 35, 38, 39, 41, 45, 50, 50, 51, 65, 70, 72, 75, 75, 82, 83, 87, 88, 90], [91, 95, 100], [102, 104, 106, 109, 112, 120, 121, 133, 134, 136, 137, 141–143, 145], [146, 151, 157, 159, 160, 165–171, 174, 175], [177–179, 181, 187, 188, 190, 196, 197, 200, 201, 203, 206, 207, 211, 214, 218, 223], [234, 236, 239, 243, 245, 248, 252, 254, 258, 275, 277–279], [284, 286–292, 294, 297, 300], [305, 306, 309–343]
Wireless, radio, antenna, signal	[14, 18, 28, 32, 33, 38, 41, 45, 46, 51, 64, 68, 70, 72, 82, 83, 85, 86, 91, 94, 96, 100], [102, 104, 106–109, 128, 131, 135, 136, 140, 141, 145, 151, 158, 160], [164, 166, 168, 170, 173, 178, 179, 188, 191, 198, 200, 201, 203, 206, 207, 209, 215–218, 223, 230], [234–236, 239, 243–245, 248, 248], [251, 252, 254, 255, 257, 272, 275, 277, 279, 284, 286, 288, 293, 299, 300, 302, 304, 305, 307–309], [311, 312, 315–318, 321, 323, 328–336, 340, 341, 343–351]
Vehicle, unmanned aerial vehicle, drone, aircraft	[10, 68, 79, 90, 91, 151, 161, 170, 181, 183, 197, 217, 220, 223, 228, 229, 232, 244, 257, 259, 266], [272, 277, 282, 299, 305, 309–311, 313, 320, 325, 336, 341, 352–354]
Cyber-physical system, robot	[15, 21, 43, 50, 56, 66, 69, 73, 74, 107, 109, 112, 122, 131, 149, 188, 189, 209, 224, 225, 229, 233], [238, 241, 242, 247–249, 258, 264, 266, 268, 282, 288, 305, 350, 354–359]
Manufacturing, factory	[19, 36, 69, 79, 93, 99, 123, 128, 149, 205, 209, 228, 247, 279, 347, 359–362]
City, building	[18, 31, 45, 66, 140, 174, 201, 204, 213, 235, 244, 282, 292, 332, 340, 355, 363–368]

systems by using a neural network embedded reinforcement learning algorithm [176].

Determining optimal sampling time for IoT devices for energy harvesting by saving batteries. Hence state space contains continuous quantities, a linear function approximation was used and a set of novel features were introduced to represent the large state space [349].

A bio-inspired RL modular architecture is able to perform skill-to-skill knowledge transfer and called transfer expert RL (TERL) model. Its architecture is based on a RL actor-critic model where both the actor and critic have a hierarchical structure, inspired by the mixture-of-experts model [392].

Deep reinforcement learning-based cooperative edge caching approach [338].

Multiple IoT devices are sending data parallel, but in general, they do not provide additional information to the existing knowledge. So, it is not necessary to permanently send data. By using actor-critic method, it can be determined which data packages need to be sent to prevent redundant or irrelevant communication [221].

Mobile edge computing and energy harvesting framework of centralized training with decentralized execution by adopting MD-hybrid-AC method [120].

Asynchronous advantage actor-critic method for mobile edge computing because computation offloading cannot have good performance in many situations, but the optimal algorithm can be chosen to use on IoT side [196].

Optimization of the robustness of IoT network topology with a scale-free network model which has good performance in random attacks. A deep deterministic learning policy (DDLP) is proposed to improve the stability for large-scale IoT applications [337].

IoT devices have lack of storage capacity, therefore a jointly cache content placement and delivery policy for the cache-enabled D2D networks was constructed. [17].

A federated reinforcement learning architecture was presented where each agent working on its independent IoT device shares its learning experience (i.e., the gradient of loss function) with each other [237].

By applying multiagent methods, there are multiple ways to organize learning:

Local learning and no centralized knowledge (see Figure 6(a))

Local knowledge deployment, local learning, and central knowledge collection

Local knowledge deployment and local learning with knowledge transfer to close neighborhoods (see Figure 6(b))

Local knowledge deployment and centralized learning (see Figure 6(c))

3.3.1. Centralized and Federated Methods. As Internet of things (IoT) services and applications are growing rapidly, most of the current optimization-based methods lack a self-adaptive ability in dynamic environments. To handle these challenges, learning-based approaches are implemented generally in a centralized way. However, network resources may be over-consumed during the training and data transmission process. To solve the complex and dynamic control issues, a federated deep reinforcement learning-based

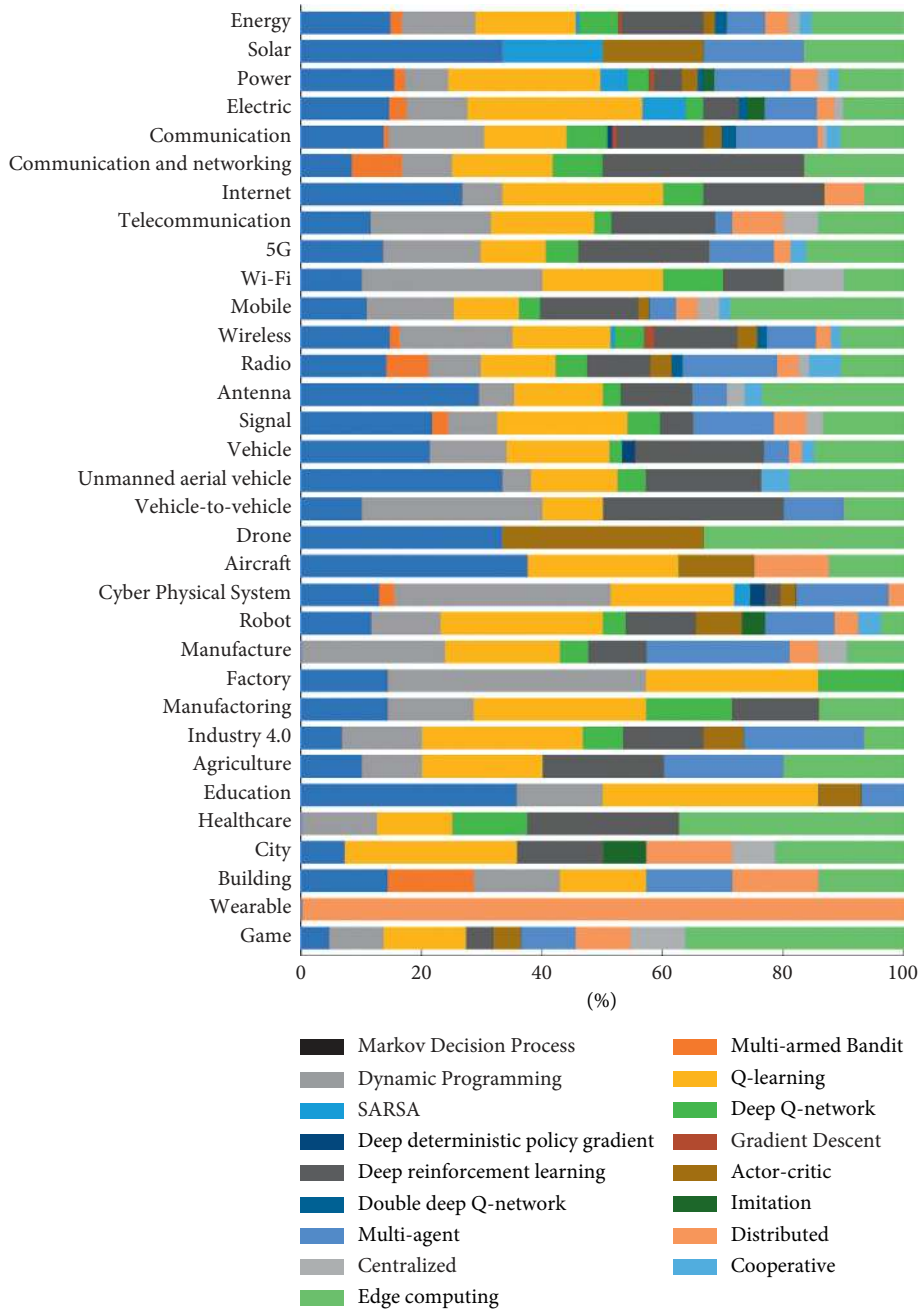

Figure 5: Distribution of RL methods by industrial field classes.

cooperative edge caching (FADE) framework is presented. FADE enables base stations (BSs) to cooperatively learn a shared predictive model by considering the first-round training parameters of the BSs as the initial input of the local training and then uploads near-optimal local parameters to the BSs to participate in the next round of global training [16].

Although the first researches have focused on designing learning algorithms with provable convergence time, but other issues, such as incentive mechanism, were explored later: a deep reinforcement learning-based incentive mechanism has been designed to determine the optimal pricing strategy for the parameter server and the optimal training strategies for edge nodes [147].

3.3.2. Hierarchical Methods. Hierarchical approaches are applied primarily to solve communication channel or information processing capacity issues. The model structure usually follows the structure of the information path. In a two-layer approach, a local IoT device needs to transfer information to a local hub and then the local hub transmits the collected information to the central decision maker. In this case, separated models can be set up for both layers to find optimal scheduling order for communication.

A new crowd sensing framework is introduced based on hierarchical structure to organize different resources and it is solved by using deep reinforcement learning-based strategy to ensure quality of service [88]. A hierarchical

Table 4: χ^2-test table of industrial field classes by RL method types.

Principle captured	Markov decision process	Multiarmed bandit	Dynamic	Temporal difference	Value function approximation	Policy gradient	Multiagent	Edge computing
Energy, solar, power, electric	0.52	1.7	−12.77	21.87	0.94	−11.54	2.05	−2.77
Communication, networking, internet, 5G, Wi-Fi, mobile	−8.61	−3.51	6.29	−18.79	2.23	13.24	−3.14	12.29
Wireless, radio, antenna, signal	5.77	3.07	−1.73	−6.62	1.33	−1.76	2.68	−2.73
Vehicle, unmanned aerial vehicle, drone, aircraft	8.44	−1.39	−2.32	−2.89	−2.49	5.9	−6.94	1.68
Cyber-physical system, robot	−1.75	0	8.14	3.86	−2.23	−2.14	1.97	−7.86
Manufacturing, factory	−3.33	−0.55	4.16	1.36	1.23	−2.54	1.52	−1.84
City, building	−1.05	0.69	−1.77	1.21	−1.01	−1.17	1.87	1.23

Table 5: Publication reference by methodological approaches.

Approach	Referred publications
Markov decision process	[12, 23, 24, 37, 64, 70, 75, 84, 96, 100, 101, 104, 127, 130, 133, 138, 144, 153, 165, 167, 170, 177, 188, 191, 199], [203, 207, 211, 212, 214, 217, 220, 231, 252, 256–259, 263, 264, 272, 274, 281, 291, 309, 313, 320, 340, 343, 346], [369–376]
Multiarmed bandit	[61, 66, 102, 198, 351, 377, 378]
Dynamic programming	[16, 19, 27, 52, 68, 70, 84, 90, 93, 107, 119, 120, 132, 135, 141, 145, 155, 156, 161, 162, 189, 191, 198, 201, 207], [209, 212, 222, 236, 242, 247, 254, 258, 259, 278, 280, 288, 289, 304, 306, 313, 321, 331, 340, 347, 357, 371, 372, 379, 380]
Q-learning	[10, 17, 24, 44, 47, 50, 64, 68, 70, 80, 81, 83, 91, 92, 94, 101, 110, 116, 124, 125, 127, 129, 133, 145, 152, 172, 179], [180, 183, 187, 201, 203, 205, 206, 208, 210, 212, 215, 219, 222–225, 227, 231, 242, 244, 246, 248, 250, 254, 262], [264, 280, 282, 283, 291, 294–296, 321, 326, 327, 343, 347, 353, 356, 366, 367, 372, 374, 381–386]
SARSA	[14, 127, 240, 246, 280, 384]
Deep Q-network	[17, 47, 83, 99, 125, 133, 190, 210, 254, 291, 294, 347, 387]
Deep deterministic policy gradient	[229, 260, 338]
Gradient descent	[26, 28, 216, 388]
Deep reinforcement learning	[19, 20, 32, 41, 47, 50, 60, 75, 77, 84, 88, 90, 95, 98, 100, 103, 117, 131, 134, 147, 154, 159, 165, 168, 176, 179, 182], [193, 199, 201, 207, 210, 220, 221, 223, 236, 241, 260, 261, 273, 275, 281, 294, 299, 301, 302, 305, 309, 311, 317, 323, 333, 338, 341, 346, 352, 355, 361, 363, 375, 380, 389–391]
Actor-critic	[15, 17, 33, 97, 120, 176, 196, 221, 237, 337, 338, 349, 392]
Double deep Q-network	[47, 83, 125, 210, 254, 294, 387, 393]
Imitation	[226, 265, 355]
Multiagent	[32, 60, 70, 77, 103, 145, 163, 168, 173, 175, 176, 188, 195, 200, 209, 218, 219, 225, 231], [245, 251, 263, 267, 280, 289, 323, 330, 338, 344, 361, 367, 377, 394, 395]
Distributed	[45, 56, 60, 73, 91, 119, 133, 145, 187, 261, 282, 348, 394]
Centralized	[60, 147, 187, 243, 296]
Cooperative	[16, 81, 170, 200, 338, 344]
Collaborative	[45, 137, 174, 196, 237, 248, 325, 381, 396]

correlated Q-learning (HCEQ) approach is presented to solve the dynamic optimization of generation command dispatch (GCD) for automatic generation control (AGC) [231]. An enhanced version of a bio-inspired reinforcement learning modular architecture is presented to perform skill-to-skill knowledge transfer and called transfer expert RL (TERL) model. TERL architecture is based on a RL actor-critic model where both the actor and critic have a hierarchical structure, inspired by the mixture-of-experts model, formed by a gating network that selects experts specializing in learning the policies or value functions of different tasks [392]. A new cloud computing model is proposed that is hierarchically composed of two layers: a cloud control layer (CCL) and a user control layer (UCL). The CCL manages cloud resource allocation, service scheduling, service profile, and service adaptation policy from a system performance point of view. Meanwhile, the UCL manages end-to-end service connection and service context from a user performance point of view. The proposed model can support nonuniform service binding and its real-time adaptation using metaobjects by intelligent service-context management using a supervised and reinforcement learning-based machine learning framework [150]. A new cooperative resource allocation algorithm is

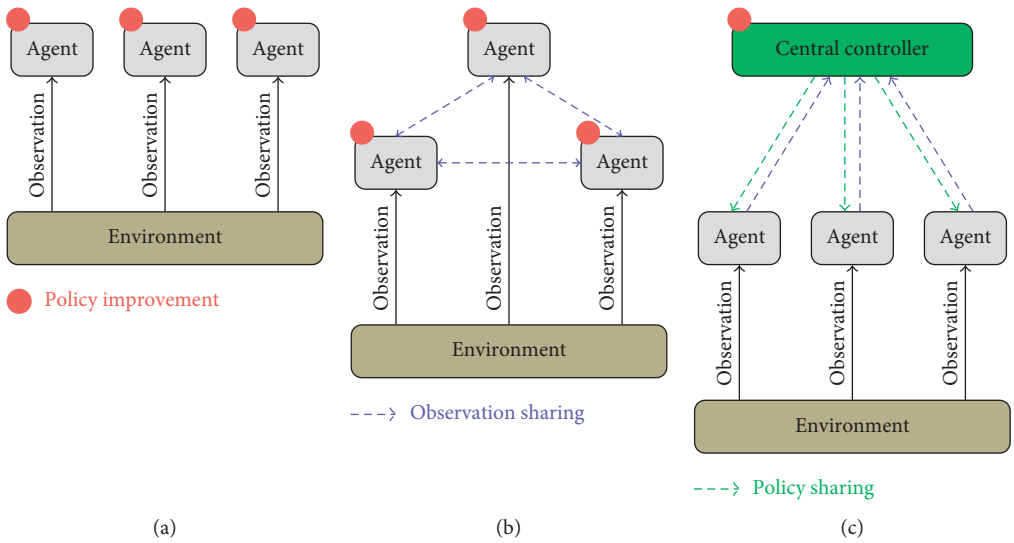

Figure 6: Cooperation concepts of multiagent learning systems. (a) Full local learning. (b) Knowledge sharing. (c) Centralized learning.

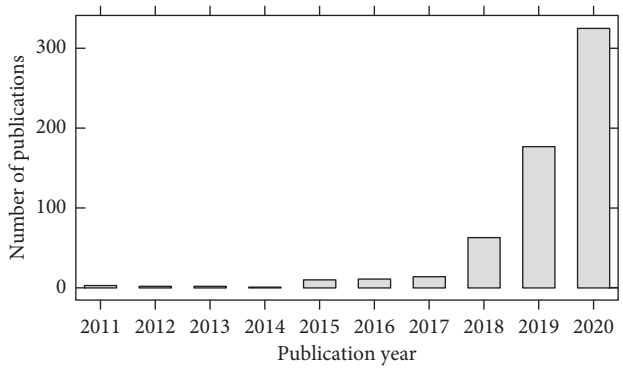

Figure 7: Distribution of publications processed in our analysis by publication years.

presented which couples reinforcement learning networks and prediction neural networks for accurate mobile targets tracking. Specifically, a hierarchical structure that performs collaborative computing is designed for alleviating computing pressure of front-end devices which are supported by edge servers [397]. A slightly different approach is applied at a resilient control problem studied for cyber-physical systems (CPSs) under the denial-of-service (DoS) attack. The term resilience is interpreted as the ability to be robust to the physical layer external disturbance and defending against cyber layer DoS attacks. The overall resilient control system is described by a hierarchical game, where the cyber security issue is modeled as a zero-sum matrix game, and physical minimax control problem is described by a zero-sum dynamic game. In virtue of the reinforcement learning method, the defense/attack policy in the cyber layer can be obtained, and additionally, the physical layer control strategy can be obtained by using the dynamical programming method [398]. Further publications in hierarchical RL topics are related to balancing timeliness and criticality when gathering data from multiple sources [116], ubiquitous user connectivity, and collaborative computation offloading for smart cities [248].

3.3.3. Distributed and Parallel Methods. It can be stated with certainty that the biggest potential of industrial applications is in intelligent devices. In this context, intelligence means some kind ability for taking autonomously decisions and furthermore being able to perform learning steps locally. There were made significant efforts to develop functional solution to reach this goal.

Computation offloading can provide a solution for the issue of the high computation requirement of resource-constrained mobile devices. The mobile cloud is the well-known existing offloading platform, which is usually far-end network solution, but this can cause other issues, such as higher latency or network delay, which negatively affects the real-time mobile Internet of things (IoT) applications. Therefore, a deep Q-learning-based autonomic management framework is proposed as a near-end network solution of computation offloading in mobile edge [133].

Another way to extend single reinforcement learning applications is to handle multiple objectives. There are two major solution practices to handle such kind of problems. The most obvious idea is to construct a mixed reward function that returns a combined result according to the different objectives [161, 259, 370]. Another possible way is to combine multiobjective ant colony optimization methods

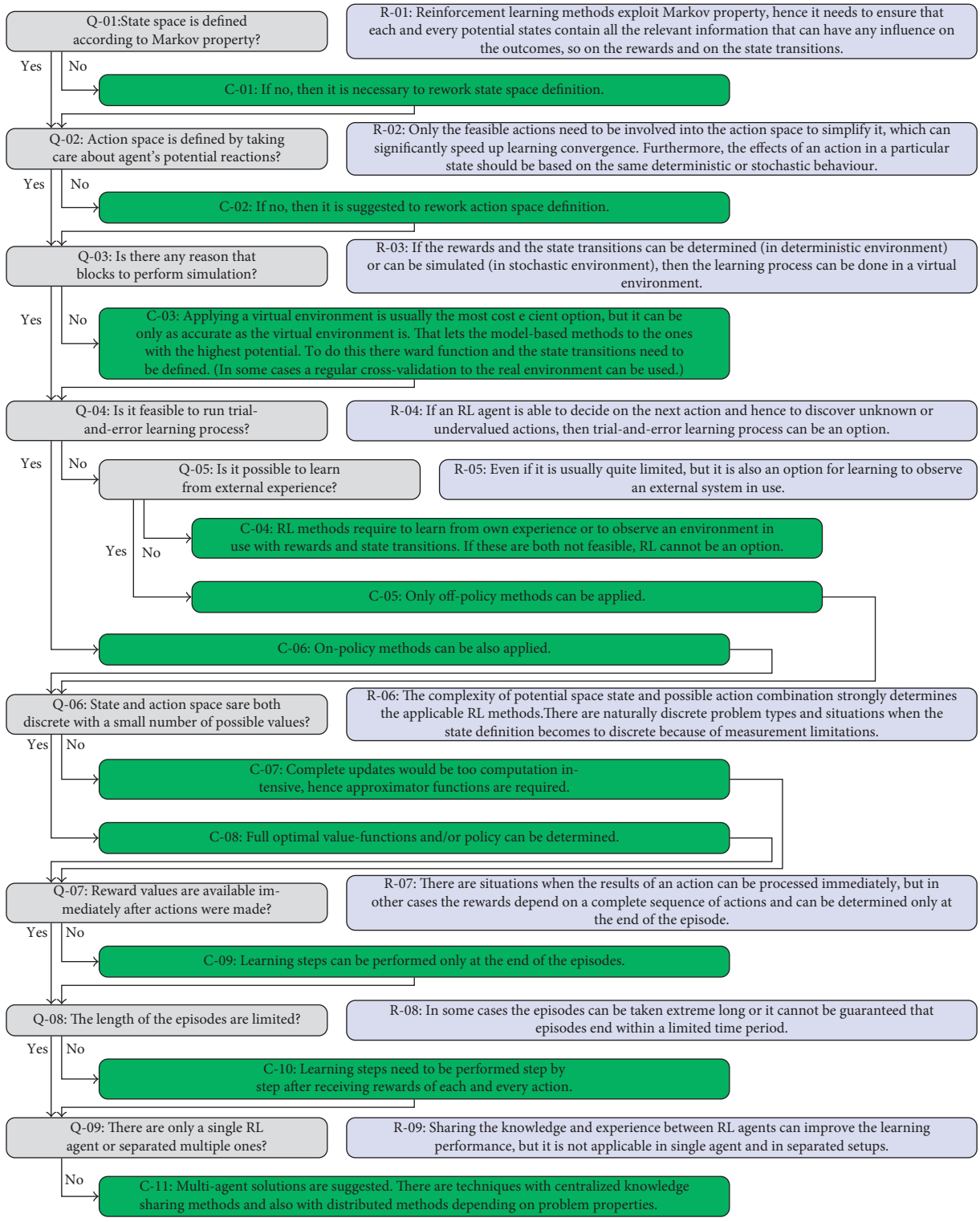

Figure 8: Guideline process to determine appropriate RL method to use.

with RL techniques like deep reinforcement learning or double Q-learning algorithms [83, 142].

3.4. General Trends of RL Applications. Before the beginning of Industry 4.0 revolution, the general methodology was based on centralized data collection, data processing, and predictive model development solutions. By spreading Internet of things (IoT) devices, it turns possible to delegate more computational task to them. This kind of potential gets being exploited by reacting to another major issue which is the lack of communication capability. On the one hand, the

communication between IoT devices and central servers or nodes are relative energy intensive processes; on the other hand, there are significant limitations on communication channels or frequencies.

By distributing computational tasks to IoT devices, a fundamental change gets required: it is not possible to assign as much human effort to data processing and predictive model development supervision as before during the centralized era. This was the major reason of appreciating RL methods because it provides a general self-learning framework that basically requires no manual or human interactions to maintain.

The early researches focused on the applicability of reinforcement learning techniques with single agents. Then, more and more complex problems were solved, and the multiagent solutions started to analyze. In the last years, the focus of the researchers is shifting to multiagent structures. The set-up of the agents and their goals or reward functions are showing very creative solutions. At a new wave of researches, the agents are defined with different roles often with attacker-defender objectives and let each of the agent to be trained an optimal strategy according to it. Then, the stability and robustness of the system can be analyzed and the weakest items can be purposefully improved.

As Figure 7 demonstrates, the number of Industry 4.0-related reinforcement learning-based researches dynamically increases, and there is no sign for expecting a slowing in it.

3.5. Discussion and Guideline Process to Determine Appropriate RL Method to Use. On the basis of the previous section, it can be highlighted that there are several ways and methods how reinforcement learning can be applied for Industry 4.0-related problems, and it is far not trivial which one can provide a successful solution.

We prepared a questionnaire and we presented it in a decision flow diagram in Figure 8. Our primary goal was to set up a method to help the readers in formulating their RL tasks. The first questions of the questionary-based process verify whether state and action spaces are appropriately defined and how the reward can be obtained. The further questions systematically narrow down the set of applicable RL methods. The possibility of using simulation or learning from own experience can determine the general learning mechanism. In contrast, the nature of reward propagation can determine a smaller subset of the RL methods that can be applicable. Even if the conclusions are soft-defined, a user with some basic knowledge of RL methods can easily interpret them, or it can be a basis of some RL methods selector wizard. We believe that researchers will have fewer failed attempts by using our guideline, and the time-to-solution can be reduced significantly.

We should keep in mind that the whole reinforcement learning concept is based on Markov decision processes. A direct conclusion is that the state space should be constructed in a way that all the potential states should contain all the relevant information that can have any influence on the outcomes. Moreover, the action space should be constructed similarly: the effects of an action in a particular state should be based on the same deterministic or stochastic behaviour. This will let the RL agent to learn the effect mechanism behind.

Once the state and action spaces are defined, it needs to be investigated whether performing simulations is an option or not. If we are able to determine the environment's behaviour when an action is made in a particular state, so deriving the reward value and the state transition, then an extensive learning process can be executed by using model-based RL methods in a cost-efficient way without significant risk of applying untrained agents. The general rule is also true in this case: the RL solution will be as adequate as the simulation is. If there is an option to validate the simulation outcomes to the real environment, then this can help to ensure the validity of the solution.

4. Conclusions

As we pointed out that reinforcement learning methods have a high potential also in Industry 4.0 applications which is a common agreement of researchers, one of the biggest reasons behind is that smart tools require a high level of optimizations which cannot be satisfied with human interventions. This continuously raises the demand of self-learning solutions, and RL techniques have been proven their efficiency at multiple fields. A major goal of our article was to give an overview of RL applications at the field of Industry 4.0. As a first step, we served a high-level overview of the general RL framework and a classification of RL methods to easily see through the possibilities, while we also presented a more detailed summary of the most widely used RL methods of Industry 4.0 applications in Appendix. Therefore, our publication can serve a starting point of further researches for RL applications.

Then, we highlighted the results of our systematic literature overview of reinforcement learning applications at the field of Industry 4.0. An extensive keyword analysis drove us to identify some typical patterns by choosing an adequate RL method for some particular combinations of principal captures and industrial fields. Although there are no unique optimal RL methods, there are RL methods that provide efficient solution for some problems. Our summary can be used as a hands-on-reference for further researches and it can help researchers to shorten the preparation time for their researches.

Furthermore, we prepared a questionnaire that provides a methodology to set up the reinforcement learning system in a proper way and to choose an appropriate method for the learning problem that the researcher is facing to. We believe that an extension of our questionnaire can be a basis of a wizard tool that enables the user to find the most fitting RL method for the learning task and guiding through the set-up processes. On the other hand, by knowing the key properties of the different RL methods, it becomes faster to adopt an existing one or to modify it to fit the specific needs and hence develop an own RL method.

We hope that our article lets the researchers strengthen to decide using RL methods for further applications as numerous successful applications show the high efficiency of them.

Appendix

In Appendix, we will describe one by one the major methods of reinforcement learning by highlighting their properties

and evolutionary stages by following David Silver's approach from the simplest ones to the more complex ones.

A. Dynamic Programming

Dynamic programming (DP) covers a decision process by breaking it down into a sequence of elementary decision steps over time. "Dynamic" refers to the sequential approach, while "programming" refers to its optimization objective.

In this section, all the methods work with the assumption that the environment is perfectly known. Iterative policy evaluation method is described for learning state-value function of a given policy Π, then value iteration method is used to determine optimal state-value function although actions are taken according to any given policy Π, and last but not least, policy iteration is presented to derive an optimal policy to the environment.

In general, there is limited usage of dynamic programming algorithms both because of its assumption to know the environment perfectly and its high computational requirements. On the other hand, dynamic programming methods provide the essence of ideas that are used in advanced methods in an easily understandable form.

Iterative Policy Evaluation. Let us assume that a policy π is given and actions are taken according to it. The goal is to determine state-value function v_π by iterative application of Bellman backup: $v_1 \longrightarrow v_2 \longrightarrow \ldots \longrightarrow v_\pi$. At each and every iteration steps, the state-value function should be updated in the following way:

$$v_{k+1}(s) = \sum_{a \in \mathscr{A}} \pi(a \mid s) \left(\mathscr{R}_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathrm{P}_{ss'}^a v_k(s') \right). \tag{A.1}$$

The second term shows the cumulative rewards from state s by taking action a and applying a single Bellman decomposition while the first term provides the probability of taking action a by following policy Π. It can be proven that with weak conditions, the proposed state-value function update will converge to $v_\pi(S)$ ([4], Section 4.2).

Value Iteration. Iterative policy evaluation method can be extended to find an optimal state-value function $v^*(s)$. The main idea behind that iteration should be done by starting from the final reward and working backward. Let us assume that the solution of subproblem $v^*(s')$ is known. Then, by the solution of the next iteration step, $v^*(s)$ can be found by one-step look-ahead:

$$v^*(s) \leftarrow \max_{a \in \mathscr{A}} \left(\mathscr{R}_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathrm{P}_{ss'}^a v^*(s') \right). \tag{A.2}$$

It can easily be seen that for finite state space $\mathscr{S}$, the determination of optimal state-value function for all the available states can be done in finite number of steps ([4], Section 4.4).

Policy Iteration. The iteratively learnt knowledge can be extracted by improving the policy by acting greedily with respect to v_{π^*}. This practically means to pick that action a from a particular state s which maximizes the sum of immediate reward r_s^a and discounted state-value $\gamma v_{\pi^*}(s')$ of the successor state s' ([4], Section 4.6). The learning process of policy iteration is demonstrated on Figure 9.

B. Model-Free Prediction Methods

Unlike in dynamic programming, in model-free methods, perfectly known environment is not necessary, only experience samples are required or with other words just sequences of states, actions, and rewards, no prior knowledge of the environment.

In this section, Monte-Carlo learning method is presented for learning simply by averaging the experience, and then temporal-difference learning method is discussed to let the agent learn by more frequent but smaller steps by applying bootstrapping techniques, while temporal-difference (λ) learning method is described as an extension of temporal-difference method's one-step learning to multiple-steps learning.

Monte-Carlo Learning. Monte-Carlo (MC) agent solves the reinforcement learning problem by applying average sample return, so it learns from complete episodes. Hence, it needs to be guaranteed always to terminate episodes; otherwise, the learning process cannot be performed. MC uses the simplest idea by assigning empirical mean of returns to a specific state ([4], Section 5.1). There are two major types of MC methods:

First-visit MC: only the first visit of a state will be involved into the calculation during an episode. Let us assume that state s is visited first time at time period t. Let us denote G_t as the total return from time period t and $N(s)$ the number of times that state s is visited while $S(s)$ is the sum of G_t returns up to the current episode. In this case, the state-value estimate will be the empirical mean: $V(s) = S(s)/N(s)$. As experience grows, so as $N(s) \longrightarrow \infty$, the long-term mean will converge to the state-value function: $V(s) \longrightarrow v_\pi(s)$.

Every-visit MC: all the visits of a state will be involved into the calculation during an episode. Formally, the main difference to first-visit MC is that $N(s)$ needs to be incremented at every time period t whenever state s is visited.

From computational point of view, it is important to mention that empirical mean is determined incrementally in practice. Let us denote $V^{(n)}(s)$ as the value-function estimate while $S^{(n)}(s)$ is the cumulative sum of returns after episode n, then $G_t^{(n)}$ is the total return in episode n from time period t when state s is visited and assume that state s is visited kth times overall.

$$\begin{aligned} V^{(n)}(s) &= \frac{1}{k} S^{(n)}(s) = \frac{1}{k} \sum_{i=1}^{n} S^{(i)}(s) = \frac{1}{k} \sum_{i=1}^{n-1} S^{(i)}(s) + \frac{1}{k} G_t^{(n)} \\ &== \frac{1}{k}(k-1) V^{(n-1)}(s) + \frac{1}{k} G_t^{(n)} = V^{(n-1)}(s) \\ &\quad + \frac{1}{k}\left(G_t^{(n)} - V^{(n-1)}(s) \right). \end{aligned} \tag{A.3}$$

Figure 10 demonstrates the learning process of Monte-Carlo method. As we can see, the learning step is performed at the end of an episode.

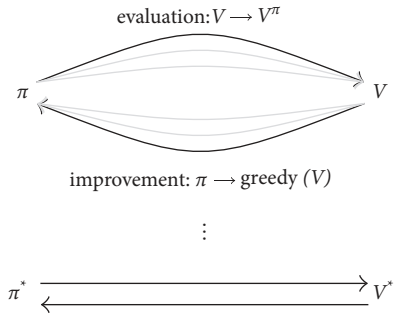

Figure 9: Learning by policy iteration method.

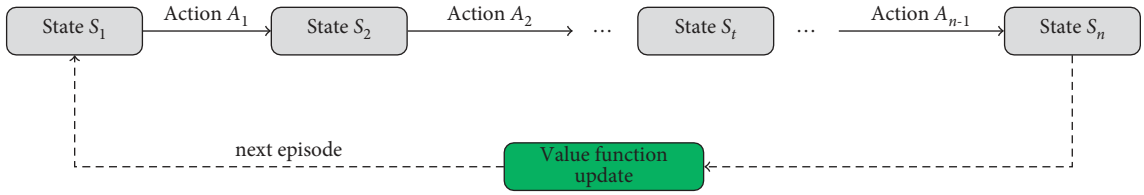

Figure 10: Monte-Carlo learning method.

Temporal-Difference Learning. Temporal-difference (TD) agent learns from incomplete episodes by applying bootstrapping. Comparing to MC learning, TD uses best guess of total return, or formally $R_{t+1} + \gamma V(S_{t+1})$ instead of episodic experience G_t to calculate value function estimates $(V(s))$. This single difference indicates that TD agent can perform a learning step after each and every actions ([4], Section 6.1), as Figure 11 shows. As a consequence, it can be applied at never ending episodes.

Temporal-Difference (λ) *Learning.* There are intermediate solutions between TD that performs VF estimate updates after 1-step return and MC that performs updates only at the end of an episode (practically ∞-step return). The main idea behind is to apply normalized geometric series $(1-\lambda)\lambda^{n-1}$ for weighting n-step returns $G_t^{(n)}$ ([4], Section 7.1). In this case, value function estimate will use a weighted total return of $G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$. It can be shown that TD(0) is equivalent to every-visit MC learning and TD(1) is equivalent to original TD learning methods. Furthermore, TD (λ) methods can be applied both forward and backward. The algorithms shown in this section can be used whether

> In offline mode: value function estimate updates are accumulated within episodes but applied only at the end of the episode, or
>
> In online mode: value function estimate updates are accumulated within episodes and can be applied immediately.

A unified view of model-free prediction techniques is shown in Figure 12. First, it was created by Richard Sutton, but this version is prepared by David Silver. It highlights the two most important dimensions of learning methods: the vertical dimension represents the depth of the updates, while the horizontal dimension represents the width of the updates.

C. Model-Free Control Methods

In the previous section, model-free prediction methods were summarized. These are methods that learn from other's experience so acting policies were managed from the external and called off-policy learning. In contrast, on-policy learning lets the algorithm to make actions on the basis of their own policy. Hence, a major objective steps to the front, to optimize policy.

In this section, ϵ-Greedy policy iteration is described to combine exploitation of the current knowledge of optimal decisions and exploration of unknown new potentials. Furthermore, on-policy temporal-difference control method known as SARSA method is presented by applying bootstrapping techniques to speed-up the learning process.

ϵ-Greedy Policy Iteration Control. ϵ-Greedy policy iteration covers a combined solution. On the one hand, MC method is applied to learn the action-value function $Q(s;a)$. On the other hand, the agent can act greedily which means that it will choose the most optimal action on the basis of the actual action-value function $Q(s;a)$. This kind of action policy exploits only the current experience and does not support to explore alternatives. With a small change in the strategy, this kind of issue can be solved: let the agent act randomly with probability ϵ and greedily with probability $(1-\epsilon)$ ([4], Section 5.4):

$$\pi(as) = \begin{cases} 1 - \epsilon + \dfrac{\epsilon}{m}, & \text{if } a = \operatorname{argmax}_{a' \in \mathcal{A}} Q(s, a'), \\ \dfrac{\epsilon}{m}, & \text{otherwise.} \end{cases} \tag{A.4}$$

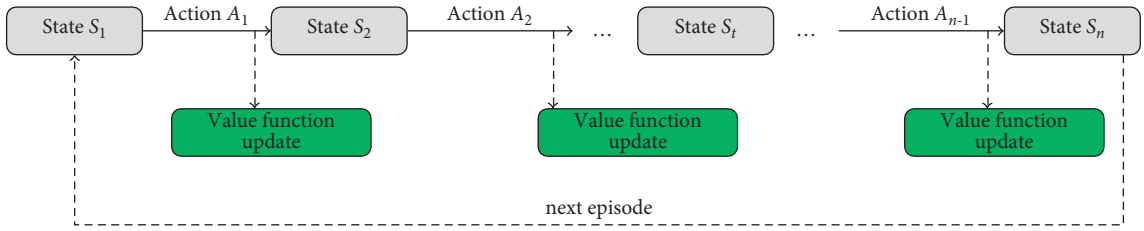

Figure 11: Temporal-difference learning method.

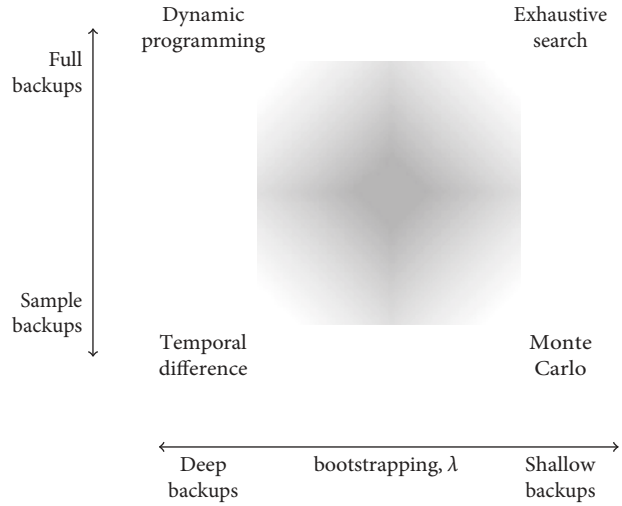

Figure 12: Unified view of model-free prediction techniques.

On-Policy Temporal-Difference Control Method, Aka SARSA Method. Similar to model-free prediction methods, there is also an algorithm to let agent learn from incomplete episodes by applying bootstrapping ([4], Section 6.4). In this case, ϵ-Greedy policy iteration method needs to be modified in the following way: instead of using MC method, TD learning should be applied for learning the action-value function $Q(s;a)$ that makes possible to perform a learning step after each and every actions and acting according to the most updated action-value function in a similar way than at ϵ-greedy policy iteration. The SARSA name comes from an acronym: state $s \longrightarrow$ action $a \longrightarrow$ reward $r \longrightarrow$ state $s' \longrightarrow$ action a'. By following SARSA method, action-value function update should look like $Q(s;a) \leftarrow Q(s;a) + \alpha(r + \gamma Q(s';a') - Q(s;a))$. It can be proved that under certain conditions, SARSA action-value function converges to optimal action-value function: $Q(s;a) \longrightarrow q^*(s;a)$.

D. Off-Policy Learning

There are several situations when the learning process is not based on just own experience. Formally, this means that target policy $\pi(a \mid s)$ or state-value function $v_\pi(s)$ or action-value function $q_\pi(s;a)$ is determined by observing results of an external behaviour policy $\mu(a \mid s)$.

In this section, importance sampling is shown to determine the most accurate of the learning objective, and then Q-learning is described as an effective alternative to get the function iteration with a lower variance.

Importance Sampling. One possible way to handle the difference of target and behaviour policy is importance sampling when a correction multiplier shall be applied by processing observations ([4], Section 5.8). If MC learning is combined with importance sampling, then value function update will look like $(S_t) \leftarrow V(S_t) + \alpha(G_t^{\pi/\mu} - V(S_t))$. But because corrections are made at the end of an episode, the product of multipliers can drive to a dramatically high variance and hence MC learning is not suitable for off-policy learning.

Therefore, TD learning seems much more adequate to combine with importance sampling, because correction multiplier should be applied for only a single step and not for a whole episode:

$$(S_t) \leftarrow V(S_t) + \alpha\left(\frac{\pi(A_t \mid S_t)}{\mu(A_t \mid S_t)}(R_{t+1} + \gamma V(S_{t+1})) - V(S_t)\right). \tag{A.5}$$

Q-Learning. Another possible way to handle the difference of target and behaviour policy is to modify the value function update logic as Q-learning does ([4], Section 6.5). Assume that in state S_t, the very next action is derived by using behaviour policy: $A_{t+1} \sim \mu(\cdot \mid S_t)$. By taking action A_{t+1}, immediate reward R_{t+1} and the next state S_{t+1} will be determined. But for value function update, let us consider an alternative successor

action on the basis of target policy: $A' \sim \pi(\cdot \mid S_t)$. Therefore, the importance sampling will be not necessary and Q-learning value function update will look like $Q(S_t; A_t) \longleftarrow Q(S_t; A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}; A') - Q(S_t; A_t))$.

In a special case, if target policy π is chosen as a pure greedy policy and behaviour policy μ follows ϵ-greedy policy, then the so-called SARSAMAX update can be defined as follows:

$Q(S; A) \leftarrow Q(S; A) + \alpha(R + \gamma \max_{a'} Q(S'; a') - Q(S; A))$.

Last but not least, it was proven that Q-learning control converges to the optimal action-value function: $Q(s; a) \longrightarrow q_*(s; a)$.

E. Value Function Approximation

The reinforcement learning methods discussed in the previous sections represented value functions by lookup tables, but in practice, it is not feasible operating with state-level or state-action-level lookup tables. On the one hand, it would be very memory- and computation-intensive, and on the other hand, the learning process would be too slow if the state and/or action spaces are large. The solution for large problems is to estimate state-value and action-value functions with function approximation: $\widehat{v}(s; \mathbf{w}) \approx v_\pi(s)$, and similarly, $\widehat{q}(s; a; \mathbf{w}) \approx q_\pi(s; a)$.

There are many kinds of function approximation methods that can be applied: linear combination of features, neural network, decision tree, and Fourier bases. In this section, the first two types of methods are discussed. The first gradient descent method is presented that can be effectively combined with Monte-Carlo or temporal-difference methods for value function approximations, and then deep Q-network is described that serves a more sample-effective way from learning.

Value Function Approximation by Gradient Descent. A well-known tool for function approximation is gradient descent ([4], Section 9.3). Let us denote $J(\mathbf{w})$ as a differentiable function of parameter vector $\mathbf{w}$. Define the gradient of $J(\mathbf{w})$ as $\nabla_{\mathbf{w}} J(\mathbf{w}) = (\partial J(\mathbf{w})/\partial w_1, \ldots, \partial J(\mathbf{w})/\partial w_n,)^T$. To find a local minimum of $J(\mathbf{w})$, parameter $\mathbf{w}$ needs to be adjusted in the direction of negative gradient by $\Delta\mathbf{w} = -1/2\alpha\nabla_{\mathbf{w}} J(\mathbf{w})$ where α is the learning step-size parameter.

An effective solution is to use gradient descent with linear combination of features, because in this case, the formulas become much simpler. Value function representation will look like $\widehat{v}(S; \mathbf{w}) = \mathbf{x}(S)^T\mathbf{w} = \sum_{i=i}^{n} x_i(S) w_i$, while objective function to minimise mean-squared error between true value function and its approximation can be calculated by the formula of $J(\mathbf{w}) = \mathbb{E}_\pi[(v_\pi(S) - \mathbf{x}(S)^T\mathbf{w})^2]$. It is proven that stochastic gradient descent with linear combination of features converges to global optimum. Furthermore, the update rule is quite simple: $\nabla_{\mathbf{w}}\widehat{v}(S; \mathbf{w}) = \mathbf{x}(S)$, and then $\Delta\mathbf{w} = \alpha(v_\pi(S) - \widehat{v}(S; \mathbf{w}))\mathbf{x}(S)$. The result shows that parameter $\mathbf{w}$ adjustment stands for three components: learning step-size, prediction error, and feature value. In practice, the true value function is usually not known but a noisy sample of it is known at different methods:

For MC method, the target is G_t and hence parameter update $\Delta\mathbf{w} = \alpha(G_t - \widehat{v}(S_t; \mathbf{w}))\nabla_{\mathbf{w}}\widehat{v}(S_t; \mathbf{w})$.

For TD(0) method, the target is the TD target $R_{t+1} + \gamma\widehat{v}(S_{t+1}; \mathbf{w})$ while parameter update

$$\Delta\mathbf{w} = \alpha\left(R_{t+1} + \gamma\widehat{v}(S_{t+1}; \mathbf{w}) - \widehat{v}(S_t; \mathbf{w})\right)\nabla_{\mathbf{w}}\widehat{v}(S_t; \mathbf{w}). \tag{A.6}$$

For TD (λ), the target is λ-return G_t^λ and parameter update $\Delta\mathbf{w} = \alpha(G_t^\lambda - \widehat{v}(S_t; \mathbf{w}))\nabla_{\mathbf{w}}\widehat{v}(S_t; \mathbf{w})$.

Whichever method is chosen, the RL learning process needs to update the value function approximation with the same frequency than at the original method.

Deep Q-Network. Even if gradient descent-based value function approximation methods can be very calculation-effective and updates can be managed incrementally, these are less sample-effective which means that the information that could be extracted from an observation will be not necessarily exploited.

There are batch methods that are working with experience replay. Preliminary all the observed experiences should be collected. Let us denote $\mathscr{D}$ as the consisting experience of state-value pairs: $\mathscr{D} = \langle\langle s_1; v_1^\pi\rangle, \ldots, \langle s_n; v_n^\pi\rangle\rangle$. Artificial observations can be generated by random sampling from experience history: $\langle s; v^\pi\rangle \sim \mathscr{D}$. Therefore, stochastic gradient descent can be applied on it: $\Delta\mathbf{w} = \alpha(v^\pi - \widehat{v}(s; \mathbf{w}))\nabla_{\mathbf{w}}\widehat{v}(s; \mathbf{w})$. In this way, $\mathbf{w}^\pi$ converges to optimal least square solution.

One of a most commonly used RL methods was born by combining experience replay and Q-learning with periodically frozen target policy:

(1) By using behaviour policy, action a_t can be taken according to ϵ-greedy policy

(2) Transitions should be stored in replay memory $\mathscr{D}$ as $\langle s_t, a_t, t_{t+1}, s_{t+1}\rangle$

(3) There can be generated random mini-batch samples of transitions (s, a, r, s') from $\mathscr{D}$

(4) On the basis of them, Q-learning targets will be determined by using fixed parameters w^-

(5) Minimise mean-squared error between Q-network and Q-learning targets:

$$\mathscr{L}_i(w_i) = \mathbb{E}_{s,a,r,s' \sim \mathscr{D}_i}\left[\left(r + \gamma\max_{a'} Q(s', a', w_i^-) - Q(s, a, w_i)\right)^2\right]. \tag{A.7}$$

F. Policy Gradient

In contrast to value-based methods where optimal action can be determined on the basis of learnt value function in a particular state, policy gradient methods approximate directly the optimal policy: $\pi_\theta(s, a) = \mathbb{P}[a \mid s, \theta]$.

It is necessary for an objective function $J(\theta)$ to measure the goodness of fitting policy π_θ to the optimal policy. In this case, policy-based RL becomes an optimization problem to

find optimal θ according to $J(\theta)$. There are methods that use gradient as gradient descent, conjugate gradient, or quasi-Newton method and there are methods that do not use as hill climbing, simplex, or genetic algorithms. In general, these kinds of methods show better convergence properties and can work effectively with high-dimensional or continuous action spaces, and last but not least, they can learn stochastic policies. On the other hand, policy gradient methods typically converge to a local rather than global optimum. It is important to highlight that value functions can be also used to learn the optimal θ parameter, but once it is learnt, value functions are not necessary to select optimal action.

Softmax. Let $J(\theta)$ be a policy objective function. Policy gradient descent algorithms search for a local optimum in $J(\theta)$ by ascending the gradient of the policy: $\Delta\theta = \alpha\nabla_\theta J(\theta)$. By assuming that policy π_θ is differentiable and its gradient is $\nabla_\theta\pi_\theta(s,a)$, likelihood ratios can be transformed to the following form: $\nabla_\theta\pi_\theta(s,a) = \pi_\theta(s,a)\nabla_\theta\pi_\theta(s,\ a)/\pi_\theta(s,a) = \pi_\theta(s,a)\nabla_\theta\log\pi_\theta(s,a)$, where $\nabla_\theta\log\pi_\theta(s,a)$ is called score function.

Softmax policy method is based on the approach of weighting actions by using linear combinations of features $\phi(s,a)^T\theta$ ([4], Section 13.2). Therefore, the probabilities of actions are proportional to exponentiated weights: $\pi_\theta(s,a) \propto e^{\phi(s,a)^T\theta}$. The score function looks like $\nabla_\theta\log\pi_\theta(s,a) = \phi(s,a) - \mathbb{E}_{\pi_\theta}[\phi(s,\cdot)]$.

Gaussian/Natural Policy Gradient. In continuous action spaces, Gaussian policy is a natural option. In this case, the mean is a linear combination of features: $\mu(s) = \phi(s,a)^T\theta$. By fixing variance as σ^2, the policy will be Gaussian: $a \sim \mathcal{N}(\mu(s), \sigma^2)$. The score function will look like $\nabla_\theta\log\pi_\theta(s,a) = 1/\sigma^2(a - \mu(s))\phi(s)$.

Monte-Carlo Policy Gradient Method Aka REINFORCE. Monte-Carlo policy gradient method or with more popular name the REINFORCE algorithm updates θ parameter by using stochastic gradient ascent. It is strongly based on-policy gradient theorem that generalizes likelihood ratio approach to multistep MDPs by replacing immediate reward r with long-term values of $Q^\pi(s,a)$ with weak restrictions on $J(\theta)$. The key idea behind that the locally optimal policy can be found by gradient ascent on the objective function as follows: $\theta_{t+1} \leftarrow \theta_t + \alpha\nabla_{\theta_t}\log\pi_{\theta_t}(s_t,a_t)v_t$, where v_t is an unbiased sample of $Q^\pi_{\theta_t}(s_t,a_t)$.

Actor-Critic Policy Gradient. In practice, REINFORCE still has high variance. To handle it, action-value function can be also estimated: $Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$. In this way, there are two sets of parameters:

Critic: it updates action-value function parameters w

Actor: it updates policy parameters θ according the actual version of critic

Updates should be done at each elementary steps as follows:

Sample reward: $r = \mathcal{R}^a_s$

Sample transition: $s' \sim \mathrm{P}^a_s$

Sample action: $a' \sim \pi_\theta(s,a')$

$\delta = r + \gamma Q_w(s',a') - Q_w(s,a)$

$\theta = \theta + \alpha\nabla_\theta\log\pi_\theta(s,a)Q_w(s,a)$

$w \longleftarrow w + \beta\gamma\phi(s,a)$

$s \longleftarrow s'$

$a \longleftarrow a'$

G. Model-Based Methods

Model-free methods learn value function and/or policy directly from their experience of a real environment. The accuracy of the knowledge of RL can be raised by extending the experience collection process. This can be reached either by setting up an artificial virtual environment due to defining reward and state transition functions that describes the real environment well or by building an own model that approximates the real environment by learning its history.

If it is assumed that the state space $\mathcal{S}$ and action space $\mathcal{A}$ are known, then model $\mathcal{M} = \langle \mathrm{P}_\eta; \mathcal{R}_\eta \rangle$ is a representation of MDP $\langle \mathcal{S}; \mathcal{A}; \mathrm{P}; \mathcal{R} \rangle$ if $S_{t+1} \sim \mathrm{P}_\eta(S_{t+1} \mid S_t, A_t)$ and $R_{t+1} = \mathcal{R}_\eta(R_{t+1} \mid S_t, A_t)$. Learning model from experience is a supervised learning problem. Figure 13 presents the basic concept of model-based learning methods.

First, the model should learn and therefore an internal simulation environment can be defined. Then, using the model representation, the model-free RL methods can be used. So, model-based techniques differ from model-free techniques by using internal model representation to derive rewards and state transitions.

H. Multiagent Learning Systems

At Industry 4.0 applications, usually not a single RL agent is set up, but multiple ones. Multiagent RL topic addresses the sequential decision-making problem of multiple autonomous agents that operate in a common or quite similar environment, each of which aims to optimize its own long-term return by interacting with the environment and a central system and/or other agents.

Markov Games. One way to generalize MDPs for applying multiple agents is Markov games (MG) or also known as stochastic games. Formally, Markov game can be defined as a tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathrm{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, N\}$ denotes the set of $N > 1$ agents, $\mathcal{S}$ denotes the state space of all the agents, and $\mathcal{A}^i$ denotes the action space of agent $i \in \mathcal{N}$. By introducing $\mathcal{A} = \mathcal{A}^1 \times \ldots \times \mathcal{A}^N$, let P: $\mathcal{S} \times \mathcal{A} \longrightarrow \mathcal{S}$ be the transition probability function from any state $s \in \mathcal{S}$ to a particular state $s' \in \mathcal{S}$ for a joint action of $a \in \mathcal{A}$, while R^i: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow \mathbb{R}$ is the reward function that determines the immediate reward by starting from state s, by taking action a and by moving to state s'. Last but not least, $\gamma \in [0,1)$ is the discount factor. Figure 14 shows the general framework of Markov games.

MG problems can be classified by knowledge sharing strategies between agents and central system and their goals: whether they can learn from each other or is it worth to share observations or policies with each other or their goals are conflicting. The main categories are

Cooperative agents problem

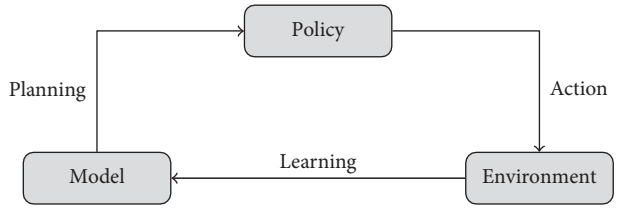

Figure 13: Model-based reinforcement learning process.

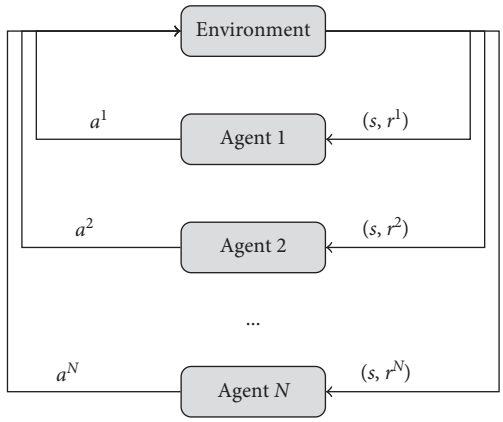

Figure 14: Schematic diagram of Markov games.

Conflicting agents problem

Mixed problem

In a fully cooperative setting, all agents have the very same or identical reward function: $R^1 = R^2 = \ldots = R^N = R$. This is also referred as multiagent MDP (MMDP). With this approach, the state- and action-value functions are identical to all agents, which thus enables the single-agent RL algorithms to be applied, if all agents are coordinated as one decision maker. The global optimum for cooperation now constitutes a Nash equilibrium of the game.

Nash equilibrium (NE) characterizes an equilibrium point π^*, from which none of the agents has any incentive to deviate. As a standard learning goal for MARL, NE always exists for discounted MGs, but may not be unique in general. Most of the MARL algorithms are contrived to converge to such an equilibrium point.

We believe that our summary of the major reinforcement learning methods gave a useful and efficient overview of the concept behind. As our literature overview shows there are numerous further modifications and extensions over the basis of the basic methods. By following our questionnaire in Figure 8, it becomes easier to determine the relevant area of RL methods that can provide an appropriate solution to be fitted to their learning problems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the TKP2020-NKA-10 project financed under the 2020-4.1.1-TKP2020 Thematic Excellence Programme by the National Research, Development and Innovation Fund of Hungary.

References

[1] Y. Lu, "Industry 4.0: a survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[2] V. Roblek, M. Meško, and A. Krapež, "A complex view of industry 4.0," *SAGE Open*, vol. 6, no. 2, 2016.

[3] J. Posada, C. Toro, I. Barandiaran et al., "Visual computing as a key enabling technology for industrie 4.0 and industrial internet," *IEEE Computer Graphics and Applications*, vol. 35, no. 2, pp. 26–40, 2015.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA, USA, 2018.

[5] R. Csalódi, Z. Süle, S. Jaskó, T. Holczinger, and J. Abonyi, "Industry 4.0-driven development of optimization algorithms: a systematic overview," *Complexity*, vol. 2021, Article ID 6621235, 22 pages, 2021.

[6] K.-D. Thoben, S. A. Wiesner, S. Wiesner, and T. Wuest, ""Industrie 4.0" and smart manufacturing - a review of research issues and application examples," *International Journal of Automation Technology*, vol. 11, no. 1, pp. 4–16, 2017.

[7] J. Mattioli, P. Perico, and P. O. Robic, *Improve Total Production Maintenance with Artificial Intelligence*, pp. 56–59, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ, USA, 2020.

[8] A. Dolgui, D. Ivanov, S. P. Sethi, and B. Sokolov, "Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications," *International Journal of Production Research*, vol. 57, no. 2, pp. 411–432, 2019.

[9] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: trends and perspectives towards Industry 4.0," *Information Fusion*, vol. 50, pp. 92–111, 2019.

[10] F. Castaño, G. Beruvides, A. Villalonga, and R. E. Haber, "Self-tuning method for increased obstacle detection reliability based on internet of things LiDAR sensor models," *Sensors*, vol. 18, no. 5, 2018.

[11] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet-of-things systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, 2019.

[12] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2009–2020, 2019.

[13] D. Nallaperuma, R. Nawaratne, T. Bandaragoda et al., "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[14] S. Shresthamali, M. Kondo, and H. Nakamura, "Adaptive power management in solar energy harvesting sensor node using reinforcement learning," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 5s, 2017.

[15] Y. P. Pane, S. P. Nageshrao, J. Kober, and R. Babuška, "Reinforcement learning based compensation methods for robot manipulators," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 236–247, 2019.

[16] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441–9455, 2020.

[17] L. Li, Y. Xu, J. Yin et al., "Deep reinforcement learning approaches for content caching in cache-enabled D2D networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 544–557, 2020.

[18] W. Hu, Y. Wen, K. Guan, G. Jin, and K. J. Tseng, "ITCM: toward learning-based thermal comfort modeling via pervasive sensing for smart buildings," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4164–4177, 2018.

[19] B. Chen, J. Wan, Y. Lan, M. Imran, D. Li, and N. Guizani, "Improving cognitive ability of edge intelligent IIoT through machine learning," *IEEE Network*, vol. 33, no. 5, pp. 61–67, 2019.

[20] C. Zhang, C. Gupta, A. Farahat, K. Ristovski, and D. Ghosh, "Equipment health indicator learning using deep reinforcement learning," *Machine Learning and Knowledge Discovery in Databases*, vol. 11053, pp. 488–504, 2019.

[21] A. Kawewong, Y. Honda, M. Tsuboyama, and O. Hasegawa, "Reasoning on the self-organizing incremental associative memory for online robot path planning," *IEICE - Transactions on Info and Systems*, vol. E93, no. 3, pp. 569–582, 2010.

[22] W. Xiong, Z. Lu, B. Li et al., "A self-adaptive approach to service deployment under mobile edge computing for autonomous driving," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 397–407, 2019.

[23] M. Chu, X. Liao, H. Li, and S. Cui, "Power control in energy harvesting multiple access system with reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9175–9186, 2019.

[24] A. Ismail and V. Cardellini, "Decentralized planning for self-adaptation in multi-cloud environment," *Communications in Computer and Information Science*, vol. 508, pp. 76–90, 2015.

[25] B. Wang, Y. Sun, T. Q. Duong, L. D. Nguyen, and L. Hanzo, "Risk-aware identification of highly suspected COVID-19 cases in social IoT: a joint graph theory and reinforcement learning approach," *IEEE Access*, vol. 8, pp. 115655–115661, 2020.

[26] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced spatiotemporal attentive graph neural networks for traffic forecasting," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6414–6428, 2020.

[27] L. Roveda, J. Maskani, P. Franceschi et al., "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent and Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[28] H. Wu, Z. Zhang, C. Jiao, C. Li, and T. Q. S. Quek, "Learn to sense: a meta-learning-based sensing and fusion framework for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8215–8227, 2019.

[29] G. Vallathan, A. John, C. Thirumalai, S. Mohan, G. Srivastava, and J. C.-W. Lin, "Suspicious activity detection using deep learning in secure assisted living IoT environments," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3242–3260, 2021.

[30] S. Chesney, K. Roy, and S. Khorsandroo, "Machine learning algorithms for preventing IoT cybersecurity attacks," *Advances in Intelligent Systems and Computing*, vol. 1252, pp. 679–686, 2021.

[31] K. H. K. Reddy, A. K. Luhach, B. Pradhan, J. K. Dash, and D. S. Roy, "A genetic algorithm for energy efficient fog layer resource management in context-aware smart cities," *Sustainable Cities and Society*, p. 63, 2020.

[32] M. S. Munir, S. F. Abedin, N. H. Tran, Z. Han, E. Huh, and C. S. Hong, "Risk-aware energy scheduling for edge computing with microgrid: a multi-agent deep reinforcement learning approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3476–3497, 2021.

[33] B. Wang, Y. Sun, M. Sun, and X. Xu, "Game-Theoretic actor-critic-based intrusion response scheme (GTAC-IRS) for wireless SDN-based IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1830–1845, 2021.

[34] E. Baccour, A. Erbad, A. Mohamed et al., "RL-OPRA: reinforcement learning for online and proactive resource allocation of crowdsourced live videos," *Future Generation Computer Systems*, vol. 112, pp. 982–995, 2020.

[35] S.-G. Choi and S.-B. Cho, "Bayesian networks + reinforcement learning: controlling group emotion from sensory stimuli," *Neurocomputing*, vol. 391, pp. 355–364, 2020.

[36] K. Lepenioti, M. Pertselakis, A. Bousdekis, A. Louca, F. Lampathaki, and D. Apostolou, "Machine learning for predictive and prescriptive analytics of operational data in smart manufacturing," *Lecture Notes in Business Information Processing*, vol. 382, 2020.

[37] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, and K. Zheng, "Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7938–7953, 2021.

[38] Z.-Y. Wu, M. Ismail, E. Serpedin, and J. Wang, "Data-driven link assignment with QoS guarantee in mobile RF-optical HetNet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5088–5102, 2020.

[39] S. Sun, X. Li, M. Liu, B. Yang, and X. Guo, "DNN inference acceleration via heterogeneous IoT devices collaboration," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 57, no. 4, pp. 709–722, 2020.

[40] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: how do IoT devices use AI to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[41] W. Liang, W. Huang, J. Long, K. Zhang, K.-C. Li, and D. Zhang, "Deep reinforcement learning for resource protection and real-time detection in IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6392–6401, 2020.

[42] X. Zhou, W. Liang, K. I.-K. Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep-learning-enhanced human activity recognition for internet of healthcare things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

[43] F. Castaño, S. Okczak, A. Villalonga, R. E. Haber, and J. Kossakowska, "Sensor reliability in cyber-physical systems using internet-of-things data: a review and case study," *Remote Sensing*, vol. 11, no. 19, 2019.

[44] S. Tu, M. Waqas, S. U. Rehman et al., "Security in fog computing: a novel technique to tackle an impersonation attack," *IEEE Access*, vol. 6, pp. 74993–75001, 2018.

[45] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained IoT

nodes: opportunities and challenges," *IEEE Design & Test*, vol. 36, no. 2, pp. 7–40, 2019.

[46] N. Aihara, K. Adachi, O. Takyu, M. Ohta, and T. Fujii, "Q-learning aided resource allocation and environment recognition in LoRaWAN with CSMA/CA," *IEEE Access*, vol. 7, pp. 152126–152137, 2019.

[47] W. Seok and C. Park, "Recognition of human motion with deep reinforcement learning," *IEIE Transactions on Smart Processing & Computing*, vol. 7, no. 3, pp. 245–250, 2018.

[48] Q. Hu, S. Lv, Z. Shi, L. Sun, and L. Xiao, "Defense against advanced persistent threats with expert system for internet of things," *Wireless Algorithms, Systems, and Applications*, vol. 10251, pp. 326–337, 2017.

[49] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the internet of things: a survey on the challenges and solutions," *Electronics (Switzerland)*, vol. 9, no. 3, 2020.

[50] R. S. Alonso, I. Sittón-Candanedo, R. Casado-Vara, J. Prieto, and J. M. Corchado, "Deep reinforcement learning for the management of software-defined networks and network function virtualization in an Edge-IoT architecture," *Sustainability*, vol. 12, no. 14, 2020.

[51] Y. Meng, S. Tu, J. Yu, and F. Huang, "Intelligent attack defense scheme based on DQL algorithm in mobile fog computing," *Journal of Visual Communication and Image Representation*, vol. 65, 2019.

[52] X. Ma and W. Shi, "AESMOTE: adversarial reinforcement learning with SMOTE for anomaly detection," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 943–956, 2020.

[53] Y. Liu, K. F. Tong, and K. K. Wong, "Reinforcement learning based routing for energy sensitive wireless mesh IoT networks," *Electronics Letters*, vol. 55, no. 17, pp. 966–968, 2019.

[54] Y. Huang, X. Guan, H. Chen, Y. Liang, S. Yuan, and T. Ohtsuki, "Risk assessment of private information inference for motion sensor embedded IoT devices," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 265–275, 2020.

[55] P. Saha and S. Mukhopadhyay, "Multispectral information fusion with reinforcement learning for object tracking in IoT edge devices," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 4333–4344, 2020.

[56] P. Sun, Y. Dong, S. Yuan, and C. Wang, "Preventive control policy construction in active distribution network of cyber-physical system with reinforcement learning," *Applied Sciences*, vol. 11, no. 1, pp. 1–20, 2020.

[57] Q.-D. Ngo, H.-T. Nguyen, H.-L. Pham et al., "A graph-based approach for IoT botnet detection using reinforcement learning," *Computational Collective Intelligence*, vol. 12496, pp. 465–478, 2020.

[58] S. Sree Dharinya and E. P. Ephzibah, "Machine intelligence and automation: deep learning concepts aiding industrial applications," *EAI/Springer Innovations in Communication and Computing*, pp. 237–248, Springer, Cham, Switzerland, 2020.

[59] Y. H. Lai, Y. C. Chang, C. W. Tsai, C. H. Lin, and M. Y. Chen, "Data fusion analysis for attention-deficit hyperactivity disorder emotion recognition with thermal image and Internet of Things devices," *Software: Practice and Experience*, vol. 51, no. 3, pp. 595–606, 2021.

[60] Z. Zhang, C. Li, S. L. Peng, and X. Pei, "A new task offloading algorithm in edge computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, 2021.

[61] L. Espinosa-Leal, A. Chapman, and M. Westerlund, "Autonomous industrial management via reinforcement learning," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 6, pp. 8427–8439, 2020.

[62] S. Khan, M. Farnsworth, R. McWilliam, and J. Erkoyuncu, "On the requirements of digital twin-driven autonomous maintenance," *Annual Reviews in Control*, vol. 50, pp. 13–28, 2020.

[63] S. A. Alghamdi, "An effective strategy for fingerprint recognition based on pRAM's neural nature with data input mappings," *Proceedings of the International Congress on Information and Communication Technology*, vol. 439, pp. 623–634, 2016.

[64] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, "A deep reinforcement learning based intrusion detection system (drl-ids) for securing wireless sensor networks and internet of things," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*,vol. 317, 2020.

[65] Y. Qin, Q. Xia, Z. Xu et al., "Enabling multicast slices in edge networks," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8485–8501, 2020.

[66] R. Heartfield, G. Loukas, A. Bezemskij, and E. Panaousis, "Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1720–1735, 2021.

[67] S. Guo, Y. Qi, Y. Jin, W. Li, X. Qiu, and L. Meng, "Endogenous trusted DRL-based service function chain orchestration for IoT," *IEEE Transactions on Computers*, vol. 70, 2021.

[68] G. Wu, "UAV-Based interference source localization: a multimodal Q-learning approach," *IEEE Access*, vol. 7, pp. 137982–137991, 2019.

[69] R. K. Dhanaraj, K. Rajkumar, and U. Hariharan, "Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning," *EAI/Springer Innovations in Communication and Computing*, pp. 55–79, Springer, Cham, Switzerland, 2020.

[70] D. Yu, P. Li, Y. Chen, Y. Ma, and J. Chen, "A time-efficient multi-protocol probe scheme for fine-grain iot device identification," *Sensors*, vol. 20, no. 7, 2020.

[71] L. Chen, Y. Xu, Z. Lu, J. Wu, K. Gai, and P. C. K. Hung, "IoT microservice deployment in edge-cloud hybrid environment using reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12610–12622, 2020.

[72] K.-H. Phung, B. Lemmens, M. Goossens, A. Nowe, L. Tran, and K. Steenhaut, "Schedule-based multi-channel communication in wireless sensor networks: a complete design and performance evaluation," *Ad Hoc Networks*, vol. 26, pp. 88–102, 2015.

[73] T. Akazaki, S. Liu, Y. Yamagata, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," *Formal Methods*, vol. 10951, pp. 456–465, 2018.

[74] K. Gai, M. Qiu, M. Liu, and H. Zhao, "Smart resource allocation using reinforcement learning in content-centric cyber-physical systems," *Lecture Notes in Computer Science*, vol. 10699, 2018.

[75] F. Jazayeri, A. Shahidinejad, and M. Ghobaei-Arani, "Autonomous computation offloading and auto-scaling the in the mobile fog computing: a deep reinforcement learning-based approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 8265–8284, 2020.

[76] A. Pauna, I. Bica, F. Pop, and A. Castiglione, "On the rewards of self-adaptive IoT honeypots," *Annales des*

Telecommunications/Annals of Telecommunications, vol. 74, no. 7-8, pp. 501–515, 2019.

[77] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9895–9903, 2020.

[78] C. Shu, Z. Zhao, G. Min, J. Hu, and J. Zhang, "Deploying network functions for multiaccess edge-IoT with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9507–9516, 2020.

[79] V. Antuori, E. Hebrard, M.-J. Huguet, S. Essodaigui, and A. Nguyen, "Leveraging reinforcement learning, constraint programming and local search: a case study in car manufacturing," *Lecture Notes in Computer Science*, vol. 12333, pp. 657–672, 2020.

[80] R. Wu, J. Gong, W. Tong, and B. Fan, "Network attack path selection and evaluation based on Q-learning," *Applied Sciences*, vol. 11, no. 1, pp. 1–13, 2021.

[81] S. Redhu and R. M. Hegde, "Cooperative network model for joint mobile sink scheduling and dynamic buffer management using Q-learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1853–1864, 2020.

[82] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: the road to pareto-optimal wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020.

[83] S. Vimal, M. Khari, R. G. Crespo, L. Kalaivani, N. Dey, and M. Kaliappan, "Energy enhancement using Multiobjective Ant colony optimization with Double Q learning algorithm for IoT based cognitive radio networks," *Computer Communications*, vol. 154, pp. 481–490, 2020.

[84] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource allocation with edge computing in IoT networks via machine learning," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3415–3426, 2020.

[85] S. K. Sathya Lakshmi Preetha, R. Dhanalakshmi, and R. Kumar, "An energy efficient framework for densely distributed WSNs IoT devices based on tree based robust cluster head," *Wireless Personal Communications*, vol. 103, no. 4, pp. 3163–3180, 2018.

[86] J. Liu, D. Li, and Y. Xu, "Collaborative online edge caching with bayesian clustering in wireless networks," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1548–1560, 2020.

[87] M. K. Pandit, R. N. Mir, and M. A. Chishti, "Adaptive task scheduling in IoT using reinforcement learning," *International Journal of Intelligent Computing and Cybernetics*, vol. 13, no. 3, pp. 261–282, 2020.

[88] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Transactions on Internet Technology*, vol. 19, no. 2, 2019.

[89] S. A. Khowaja and P. Khuwaja, *Q-learning and LSTM Based Deep Active Learning Strategy for Malware Defense in Industrial IoT Applications*, Multimedia Tools and Applications, New York, NY, USA, 2021.

[90] A. Sharif, J. P. Li, M. A. Saleem et al., "A dynamic clustering technique based on deep reinforcement learning for Internet of vehicles," *Journal of Intelligent Manufacturing*, vol. 32, no. 3, pp. 757–768, 2021.

[91] F. Hussain, R. Hussain, A. Anpalagan, and A. Benslimane, "A new block-based reinforcement learning approach for distributed resource allocation in clustered IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2891–2904, 2020.

[92] K. Nivitha, A. Solaiappan, and P. Pabitha, "Robust service selection through intelligent clustering in an uncertain environment," *Intelligence in Big Data Technologies-Beyond the Hype*, vol. 1167, pp. 325–332, 2021.

[93] T. Zhou, D. Tang, H. Zhu, and L. Wang, "Reinforcement learning with composite rewards for production scheduling in a smart factory," *IEEE Access*, vol. 9, pp. 752–766, 2021.

[94] J. H. Cho and H. Lee, "Dynamic topology model of q-learning leach using disposable sensors in autonomous things environment," *Applied Sciences*, vol. 10, no. 24, pp. 1–19, 2020.

[95] H. Qi, X. Mu, and Y. Shi, "A task unloading strategy of IoT devices using deep reinforcement learning based on mobile cloud computing environment," *Wireless Networks*, 2020.

[96] X. Cheng, F. Lyu, W. Quan et al., "Space/aerial-Assisted computing offloading for IoT applications: a learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[97] Y. Wei, F. R. Yu, M. Song, and 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*, vol. 6, no. 2, pp. 2061–2073, 2019.

[98] F. Bu and X. Wang, "A smart agriculture IoT system based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 99, pp. 500–507, 2019.

[99] C.-C. Lin, D.-J. Deng, Y.-L. Chih, and H.-T. Chiu, "Smart manufacturing scheduling with edge computing using multiclass deep Q network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4276–4284, 2019.

[100] H. Yang, A. Alphones, W.-D. Zhong, C. Chen, and X. Xie, "Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5565–5576, 2020.

[101] T. Yu, B. Zhou, K. W. Chan, and E. Lu, "Stochastic optimal CPS relaxed control methodology for interconnected power systems using Q-learning method," *Journal of Energy Engineering*, vol. 137, no. 3, pp. 116–129, 2011.

[102] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-Armed bandit learning in IoT networks: learning helps even in non-stationary settings," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 228, pp. 173–185, 2018.

[103] S. Shen, Y. Han, X. Wang, and Y. Wang, "Computation offloading with multiple agents in edge-computing–supported IoT," *ACM Transactions on Sensor Networks*, vol. 16, no. 1, 2019.

[104] A. Nassar and Y. Yilmaz, "Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements," *IEEE Access*, vol. 7, pp. 128014–128025, 2019.

[105] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep reinforcement learning for autonomous internet of things: model, applications and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1722–1760, 2020.

[106] R. Ali, Y. A. Qadri, Y. Bin Zikria, T. Umer, B. S. Kim, and S. W. Kim, "Q-learning-enabled channel access in next-generation dense wireless networks for IoT-based eHealth systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[107] J. Zhang and J. Sun, "A game theoretic approach to multi-channel transmission scheduling for multiple linear systems under DoS attacks," *Systems & Control Letters*, vol. 133, 2019.

[108] M. Kwon, J. Lee, and H. Park, "Intelligent IoT connectivity: deep reinforcement learning approach," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2782–2791, 2020.

[109] M. Camelo, M. Claeys, and S. Latre, "Parallel reinforcement learning with minimal communication overhead for IoT environments," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1387–1400, 2020.

[110] P. Farhat, H. Sami, and A. Mourad, "Reinforcement R-learning model for time scheduling of on-demand fog placement," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 388–410, 2020.

[111] X. Wei, J. Zhao, L. Zhou, and Y. Qian, "Broad reinforcement learning for supporting fast autonomous IoT," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7010–7020, 2020.

[112] S. G. Choi and S. B. Cho, "Sensor information fusion by integrated AI to control public emotion in a cyber-physical environment," *Sensors*, vol. 18, no. 11, 2018.

[113] G. Rjoub, J. Bentahar, O. Abdel Wahab, and A. Saleh Bataineh, "Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems," *Concurrency Computation*, vol. 33, no. 23, 2020.

[114] P. Loreti, L. Bracciale, and G. Bianchi, "StableSENS: sampling time decision algorithm for IoT energy harvesting devices," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9908–9918, 2019.

[115] S. Shukla, M. F. Hassan, L. T. Jung, and A. Awang, "Architecture for latency reduction in healthcare internet-of-things using reinforcement learning and fuzzy based fog computing," *Advances in Intelligent Systems and Computing*, vol. 843, pp. 372–383, 2019.

[116] H. Rashtian and S. Gopalakrishnan, "Balancing message criticality and timeliness in IoT networks," *IEEE Access*, vol. 7, pp. 145738–145745, 2019.

[117] Y. Dai, G. Wang, K. Muhammad, and S. Liu, "A closed-loop healthcare processing approach based on deep reinforcement learning," *Multimedia Tools and Applications*, p. 79, 2020.

[118] C. Lork, W. T. Li, Y. Qin, Y. Zhou, C. Yuen, and W. Tushar, "An uncertainty-aware deep reinforcement learning framework for residential air conditioning energy management," *Applied Energy*, p. 276, 2020.

[119] M. Faraji Mehmandar, S. Jabbehdari, and H. Haj Seyyed Javadi, "A dynamic fog service provisioning approach for IoT applications," *International Journal of Communication Systems*, vol. 33, no. 14, 2020.

[120] J. Zhang, J. Du, Y. Shen, and J. Wang, "Dynamic computation offloading with energy harvesting devices: a hybrid-decision-based deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9303–9317, 2020.

[121] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Collaborative edge computing and caching with deep reinforcement learning decision agents," *IEEE Access*, vol. 8, pp. 120604–120612, 2020.

[122] S. Chen, J. Wang, H. Li, Z. Wang, F. Liu, and S. Li, "Top-down human-cyber-physical data fusion based on reinforcement learning," *IEEE Access*, vol. 8, pp. 134233–134245, 2020.

[123] J. Leng, G. Ruan, Y. Song, Q. Liu, Y. Fu, and K. Ding, "A loosely-coupled deep reinforcement learning approach for order acceptance decision of mass-individualized printed circuit board manufacturing in industry 4.0," *Journal of Cleaner Production*, p. 280, 2021.

[124] M. Mobasheri, Y. Kim, and W. Kim, "Fog fragment cooperation on bandwidth management based on reinforcement learning," *Sensors*, vol. 20, no. 23, pp. 1–15, 2020.

[125] G. Rjoub, O. Abdel Wahab, J. Bentahar, and A. Bataineh, "A trust and energy-aware double deep reinforcement learning scheduling strategy for federated learning on IoT devices," *Service-Oriented Computing*, vol. 12571, pp. 319–333, 2020.

[126] J. Na, H. Zhang, X. Deng, B. Zhang, and Z. Ye, "Accelerate personalized iot service provision by cloud-aided edge reinforcement learning: a case study on smart lighting," *Service-Oriented Computing*, vol. 12571, pp. 69–84, 2020.

[127] M. Tiwari, S. Misra, P. K. Bishoyi, and L. T. Yang, "Devote: criticality-aware federated service provisioning in fog-based Iot environments," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10631–10638, 2021.

[128] A. Haldorai, A. Ramu, and M. Suriya, "Organization internet of things (IoTs): supervised, unsupervised, and reinforcement learning," *EAI/Springer Innovations in Communication and Computing*, pp. 237–248, Springer, Cham, Switzerland, 2020.

[129] A. Musaddiq, R. Ali, J.-G. Choi, B.-S. Kim, and S. Won Kim, "Collision observation-based optimization of low-power and lossy IoT network using reinforcement learning," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 799–814, 2021.

[130] X. Zhou, X. Dong, Z. Laiping, K. Li, and T. Qiu, "Learning-driven cloud resource provision policy for content providers with competitors," *IEEE Transactions on Cloud Computing*, p. 8, 2020.

[131] Y. Hao, M. Chen, H. Gharavi, Y. Zhang, and K. Hwang, "Deep reinforcement learning for edge service placement in softwarized industrial cyber-physical system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5552–5561, 2021.

[132] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the internet of things: finite resources and heterogeneity," *IEEE Access*, vol. 4, pp. 7063–7073, 2016.

[133] M. G. R. Alam, M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for IoT applications," *Future Generation Computer Systems*, vol. 90, pp. 149–157, 2019.

[134] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3516–3526, 2019.

[135] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric IoT," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 781–796, 2020.

[136] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2020.

[137] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "IRAF: a deep reinforcement learning approach for collaborative mobile edge computing IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7011–7024, 2019.

[138] S. Deng, Z. Xiang, P. Zhao et al., "Dynamical resource allocation in edge for trustable internet-of-things systems: a reinforcement learning method," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6103–6113, 2020.

[139] K. Gai and M. Qiu, "Optimal resource allocation using reinforcement learning for IoT content-centric services," *Applied Soft Computing*, vol. 70, pp. 12–21, 2018.

[140] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge

computing-supported internet of things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.

[141] Y. Li, H. Ji, X. Li, and V. C. M. Leung, "Dynamic channel selection with reinforcement learning for cognitive WLAN over fiber," *International Journal of Communication Systems*, vol. 25, no. 8, pp. 1077–1090, 2012.

[142] S. Vimal, M. Khari, N. Dey, R. G. Crespo, and Y. Harold Robinson, "Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT," *Computer Communications*, vol. 151, pp. 355–364, 2020.

[143] A. Alsarhan, A. Itradat, A. Y. Al-Dubai, A. Y. Zomaya, and G. Min, "Adaptive resource allocation and provisioning in multi-service cloud environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 1, pp. 31–42, 2018.

[144] Z. Wei, B. Zhao, J. Su, and X. Lu, "Dynamic edge computation offloading for internet of things with energy harvesting: a learning method," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4436–4447, 2019.

[145] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, "Distributed Q-learning aided heterogeneous network association for energy-efficient IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2756–2764, 2020.

[146] L. Mai, N. N. Dao, and M. Park, "Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing," *Sensors*, vol. 18, no. 9, 2018.

[147] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[148] M. Khichane, P. Albert, and C. Solnon, "Strong combination of ant colony optimization with constraint programming optimization," *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, vol. 6140, pp. 232–245, 2010.

[149] A. Villalonga, G. Beruvides, F. Castano, and R. E. Haber, "Cloud-based industrial cyber-physical system for data-driven reasoning: a review and use case on an industry 4.0 pilot line," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5975–5984, 2020.

[150] T.-D. Lee, B. M. Lee, and W. Noh, "Hierarchical cloud computing architecture for context-aware IoT services," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 222–230, 2018.

[151] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6898–6908, 2020.

[152] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Dynamic service function chain embedding for NFV-enabled IoT: a deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 507–519, 2020.

[153] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[154] A. Chowdhury, S. A. Raut, and H. S. Narman, "DA-DRLS: d," *Journal of Network and Computer Applications*, vol. 138, pp. 51–65, 2019.

[155] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 102–108, 2019.

[156] F. M. Talaat, M. S. Saraya, A. I. Saleh, H. A. Ali, and S. H. Ali, "A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4951–4966, 2020.

[157] C. Qiu, H. Yao, F. R. Yu, C. Jiang, and S. Guo, "A service-oriented permissioned blockchain for the internet of things," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 203–215, 2020.

[158] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by lyapunov online reinforcement learning," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 556–565, 2020.

[159] R. Zhao, X. Wang, J. Xia, and L. Fan, "Deep reinforcement learning based mobile edge computing for intelligent Internet of Things," *Physical Communication*, vol. 43, 2020.

[160] M. Nduwayezu, Q.-V. Pham, and W.-J. Hwang, "Online computation offloading in NOMA-based multi-access edge computing: a deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 99098–99109, 2020.

[161] T. Fu, C. Wang, and N. Cheng, "Deep-learning-based joint optimization of renewable energy storage and routing in vehicular energy network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6229–6241, 2020.

[162] S. Guo, Y. Dai, S. Xu, X. Qiu, and F. Qi, "Trusted cloud-edge network resource management: DRL-driven service function chain orchestration for IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6010–6022, 2020.

[163] Y. Zhang, B. Song, Y. Zhang, X. Du, and M. Guizani, "Market model for resource allocation in emerging sensor networks with reinforcement learning," *Sensors*, vol. 16, no. 12, 2016.

[164] S. Wan, J. Lu, P. Fan, and K. B. Letaief, "Toward big data processing in IoT: path planning and resource management of UAV base stations in mobile-edge computing system," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5995–6009, 2020.

[165] G. Cui, X. Li, L. Xu, and W. Wang, "Latency and energy optimization for MEC enhanced SAT-IoT networks," *IEEE Access*, vol. 8, pp. 55915–55926, 2020.

[166] H. Yang, X. Xie, and M. Kadoch, "Machine learning techniques and A case study for intelligent wireless networks," *IEEE Network*, vol. 34, no. 3, pp. 208–215, 2020.

[167] I. Khan, X. Tao, G. M. S. Rahman, W. U. Rehman, and T. Salam, "Advanced energy-efficient computation offloading using deep reinforcement learning in MTC edge computing," *IEEE Access*, vol. 8, pp. 82867–82875, 2020.

[168] H. Yang, W.-D. Zhong, C. Chen, A. Alphones, and X. Xie, "Deep-reinforcement-learning-based energy-efficient resource management for social and cognitive internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5677–5689, 2020.

[169] Q. Li, H. Yao, T. Mai, C. Jiang, and Y. Zhang, "Reinforcement-learning- and belief-learning-based double auction mechanism for edge computing resource allocation," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5976–5985, 2020.

[170] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12229–12239, 2020.

[171] A. Ashiquzzaman, H. Lee, T.-W. Um, and J. Kim, "Energy-efficient IoT sensor calibration with deep reinforcement learning," *IEEE Access*, vol. 8, pp. 97045–97055, 2020.

[172] J. Zhang, M. Dai, and Z. Su, "Task allocation with unmanned surface vehicles in smart ocean IoT," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9702–9713, 2020.

[173] D. Wang, W. Zhang, B. Song, X. Du, and M. Guizani, "Market-based model in CR-IoT: a Q-probabilistic multi-agent reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 179–188, 2020.

[174] S. Xu, Q. Liu, B. Gong et al., "RJCC: reinforcement-learning-based joint communicational-and-computational resource allocation mechanism for smart city IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8059–8076, 2020.

[175] X. Chen and G. Liu, "Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10843–10856, 2021.

[176] Y. Zhao, L. Wang, S. Li, F. Zhou, X. Lin, and Q. Lu, "A visual analysis approach for understanding durability test data of automotive products," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, 2019.

[177] Y.-H. Xu, Y.-B. Tian, P. K. Searyoh, G. Yu, and Y.-T. Yong, "Deep reinforcement learning-based resource allocation strategy for energy harvesting-powered cognitive machine-to-machine networks," *Computer Communications*, vol. 160, pp. 706–717, 2020.

[178] N. N. Khumalo, O. O. Oyerinde, and L. Mfupe, "Reinforcement learning-based resource management model for fog radio access network architectures in 5G," *IEEE Access*, vol. 9, pp. 12706–12716, 2021.

[179] S. Ge, B. Lu, L. Xiao, J. Gong, X. Chen, and Y. Liu, "Mobile edge computing against smart attacks with deep reinforcement learning in cognitive MIMO IoT systems," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1851–1862, 2020.

[180] I. Alqerm and J. Pan, "Enhanced online Q-learning scheme for resource allocation with maximum utility and fairness in edge-IoT networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3074–3086, 2020.

[181] Q. Qi, L. Zhang, J. Wang et al., "Scalable parallel task scheduling for autonomous driving using multi-task deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13861–13874, 2020.

[182] G. Sun, R. Ou, and G. Liu, "Deep reinforcement learning-based resource reservation algorithm for emergency Internet-of-things slice," *Tongxin Xuebao/Journal on Communications*, vol. 41, no. 9, pp. 8–20, 2020.

[183] Y. Liao, X. Qiao, Q. Yu, and Q. Liu, "Intelligent dynamic service pricing strategy for multi-user vehicle-aided MEC networks," *Future Generation Computer Systems*, vol. 114, pp. 15–22, 2021.

[184] H. Qin, S. Zawad, Y. Zhou, S. Padhi, L. Yang, and F. Yan, "Reinforcement-learning-empowered MLaaS scheduling for serving intelligent internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6325–6337, 2020.

[185] S. Venticinque, S. Nacchia, and S. A. Maisto, "Reinforcement learning for resource allocation in cloud datacenter," *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, vol. 96, pp. 648–657, 2020.

[186] V. K. Prasad and M. D. Bhavsar, "Monitoring and prediction of sla for iot based cloud," *Scalable Computing: Practice and Experience*, vol. 21, no. 3, pp. 349–358, 2020.

[187] T. Liu, R. Luo, F. Xu, C. Fan, and C. Zhao, "Distributed learning based joint communication and computation strategy of iot devices in smart cities," *Sensors*, vol. 20, no. 4, 2020.

[188] I. Budhiraja, N. Kumar, and S. Tyagi, "Deep-reinforcement-learning-based proportional fair scheduling control scheme for underlay D2D communication," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3143–3156, 2021.

[189] S. Ramakrishna, C. Harstell, M. P. Burruss, G. Karsai, and A. Dubey, "Dynamic-weighted simplex strategy for learning enabled cyber physical systems," *Journal of Systems Architecture*, p. 111, 2020.

[190] M. Li, F. R. Yu, P. Si, W. Wu, and Y. Zhang, "Resource optimization for delay-tolerant data in blockchain-enabled IoT with edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9399–9412, 2020.

[191] A. Sapio, S. S. Bhattacharyya, and M. Wolf, "Runtime adaptation in wireless sensor nodes using structured learning," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 4, 2020.

[192] K. Priyadharshini and R. A. Canessane, "Light chain consensus reinforcement machine learning: an effective blockchain model for internet of things using for its advancement and challenges," *Computational Intelligence*, vol. 36, pp. 1–22, 2020.

[193] N. Yuan, C. Jia, J. Lu et al., "A DRL-based container placement scheme with auxiliary tasks," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1657–1671, 2020.

[194] M. Laroui, H. Ibn-Khedher, M. Ali Cherif, H. Moungla, H. Afifi, and A. E. Kamel, "So-vmec: service offloading in virtual mobile edge computing using deep reinforcement learning," *Transactions on Emerging Telecommunications Technologies*, vol. 32, 2021.

[195] J. Kim, D. Ryu, J. Kim, and J. H. Kim, "Two-stage hybrid network clustering using multi-agent reinforcement learning," *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–16, 2021.

[196] X. Shu, L. Wu, X. Qin, R. Yang, Y. Wu, and D. Wang, "Deep reinforcement learning cloud-edge-terminal computation resource allocation mechanism for IoT," *Advances in Intelligent Systems and Computing*, vol. 1274, 2021.

[197] J. Zhang, H. Guo, and J. Liu, "Adaptive task offloading in vehicular edge computing networks: a reinforcement learning based scheme," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1736–1745, 2020.

[198] A. P. Ortega, S. D. Ramchurn, L. Tran-Thanh, and G. V. Merrett, "Partner selection in self-organised wireless sensor networks for opportunistic energy negotiation: a multi-armed bandit based approach," *Ad Hoc Networks*, p. 112, 2021.

[199] H. Gao, Y. Xiao, H. Yan, Y. Tian, D. Wang, and W. Wang, "A learning-based credible participant recruitment strategy for mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5302–5314, 2020.

[200] H. Yang, Z. Xiong, J. Zhao, D. Niyato, C. Yuen, and R. Deng, "Deep reinforcement learning based massive access management for ultra-reliable low-latency communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2977–2990, 2021.

[201] Y. Liu, H. Lu, X. Li, Y. Zhang, L. Xi, and D. Zhao, "Dynamic service function chain orchestration for NFV/MEC-enabled iot networks: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7450–7465, 2020.

[202] W. Zhang, D. Yang, P. Haixia, W. Wu, W. Quan, and H. Zhang, "Deep reinforcement learning based resource management for dnn inference in industrial IoT," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7605–7618, 2021.

[203] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-q-learning-based transmission scheduling mechanism for the cognitive internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2375–2385, 2018.

[204] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Network*, vol. 33, no. 2, pp. 111–117, 2019.

[205] Y.-R. Shiue, K.-C. Lee, and C.-T. Su, "Real-time scheduling for a smart factory using a reinforcement learning approach," *Computers & Industrial Engineering*, vol. 125, pp. 604–614, 2018.

[206] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: current issues and machine learning-assisted solutions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 426–471, 2020.

[207] L. Lei, H. Xu, X. Xiong, K. Zheng, W. Xiang, and X. Wang, "Multiuser resource control with deep reinforcement learning in IoT edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10119–10133, 2019.

[208] J. Ge, B. Liu, T. Wang, Q. Yang, A. Liu, and A. Li, *Q-learning based flexible task scheduling in a global view for the internet of things. Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 8, 2020.

[209] Q. Tan, Y. Tong, S. Wu, and D. Li, "Modeling, planning, and scheduling of shop-floor assembly process with dynamic cyber-physical interactions: a case study for CPS-based smart industrial robot production," *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3979–3989, 2019.

[210] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," *Future Generation Computer Systems*, vol. 110, pp. 1098–1115, 2020.

[211] B. Yin, S. Zhang, and Y. Cheng, "Application-Oriented scheduling for optimizing the age of correlated information: a deep-reinforcement-learning-based approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8748–8759, 2020.

[212] D. Kim, T. Lee, S. Kim, B. Lee, and H. Y. Youn, "Adaptive packet scheduling in IoT environment based on Q-learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 6, pp. 2225–2235, 2020.

[213] S. Park, S. Park, M. I. Choi et al., "Reinforcement learning-based bems architecture for energy usage optimization," *Sensors*, vol. 20, no. 17, pp. 1–33, 2020.

[214] H. He, H. Shan, A. Huang, Q. Ye, and W. Zhuang, "Edge-Aided computing and transmission scheduling for LTE-U-enabled IoT," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7881–7896, 2020.

[215] X. Fu, L. Lopez-Estrada, and J. G. Kim, "A Q-learning-based approach for enhancing energy efficiency of bluetooth low energy," *IEEE Access*, vol. 9, pp. 21286–21295, 2021.

[216] H.-S. Lee and J.-W. Lee, "Adaptive wireless power transfer beam scheduling for non-static iot devices using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 206659–206673, 2020.

[217] M. Samir, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Online altitude control and scheduling policy for minimizing AoI in UAV-assisted IoT wireless networks," *IEEE Transactions on Mobile Computing*, p. 19, 2020.

[218] H. Park, H. Kim, S.-T. Kim, and P. Mah, "Multi-agent reinforcement-learning-based time-slotted channel hopping medium access control scheduling scheme," *IEEE Access*, vol. 8, pp. 139727–139736, 2020.

[219] Y. Martínez Jiménez, J. Coto Palacio, and A. Nowé, "Multi-agent reinforcement learning tool for job shop scheduling problems," *Communications in Computer and Information Science*, vol. 1173, 2020.

[220] H. Hu, X. Jia, Q. He, S. Fu, and K. Liu, "Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0," *Computers & Industrial Engineering*, p. 149, 2020.

[221] H. Rashtian and S. Gopalakrishnan, "Using deep reinforcement learning to improve sensor selection in the internet of things," *IEEE Access*, vol. 8, pp. 95208–95222, 2020.

[222] Z. Wang, J. Wang, F. Yang, and M. Lin, "Q-learning-based energy transmission scheduling over a fading channel," *Journal of Southeast University*, vol. 36, no. 4, pp. 393–398, 2020.

[223] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[224] T. Yu, B. Zhou, K. W. Chan, L. Chen, and B. Yang, "Stochastic optimal relaxed automatic generation control in non-markov environment based on multi-step $Q(\lambda)$ learning," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1272–1282, 2011.

[225] T. Yu, H. Z. Wang, B. Zhou, K. W. Chan, and J. Tang, "Multi-agent correlated equilibrium Q(λ) learning for coordinated smart generation control of interconnected power grids," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1669–1679, 2015.

[226] T. Yu, B. Zhou, K. W. Chan, Y. Yuan, B. Yang, and Q. H. Wu, "R(λ) imitation learning for automatic generation control of interconnected power grids," *Automatica*, vol. 48, no. 9, pp. 2130–2136, 2012.

[227] J. Xia, Y. Xu, D. Deng, Q. Zhou, and L. Fan, "Intelligent secure communication for internet of things with statistical channel state information of attacker," *IEEE Access*, vol. 7, pp. 144481–144488, 2019.

[228] J. Shao, X. Zhao, J. Yang, W. Zhang, Y. Kang, and X. Zhao, "Reinforcement learning algorithm for path following control of articulated vehicle," *Nongye Jixie Xuebao/Transactions of the Chinese Society for Agricultural Machinery*, vol. 48, no. 3, pp. 376–382, 2017.

[229] H. D. Tran, F. Cai, M. Lopez Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, "Safety verification of cyber-physical systems with reinforcement learning control," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 5s, 2019.

[230] J. Kang and D. S. Eom, "Offloading and transmission strategies for IoT edge devices and networks," *Sensors*, vol. 19, no. 4, 2019.

[231] X. Zhang, T. Yu, and J. Tang, "Optimal CPS command dispatch based on hierarchically correlated equilibrium reinforcement learning," *Dianli Xitong Zidonghua/Automation of Electric Power Systems*, vol. 39, no. 8, pp. 80–86, 2015.

[232] G. Faraci, A. Raciti, S. A. Rizzo, and G. Schembra, "Green wireless power transfer system for a drone fleet managed by reinforcement learning in smart industry," *Applied Energy*, vol. 259, 2020.

[233] J.-B. Kim, H.-K. Lim, C.-M. Kim, M.-S. Kim, Y.-G. Hong, and Y.-H. Han, "Imitation reinforcement learning-based remote rotary inverted pendulum control in openflow network," *IEEE Access*, vol. 7, pp. 36682–36690, 2019.

[234] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Energy-efficient orchestration in wireless

powered internet of things infrastructures," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 317–328, 2019.

[235] B.-N. Trinh, L. Murphy, and G.-M. Muntean, "A reinforcement learning-based duty cycle adjustment technique in wireless multimedia sensor networks," *IEEE Access*, vol. 8, pp. 58774–58787, 2020.

[236] D. Pacheco-Paramo, L. Tello-Oquendo, V. Pla, and J. Martinez-Bauset, "Deep reinforcement learning mechanism for dynamic access control in wireless networks handling mMTC," *Ad Hoc Networks*, vol. 94, 2019.

[237] H. K. Lim, J. B. Kim, J. S. Heo, and Y. H. Han, "Federated reinforcement learning for training control policies on multiple IoT devices," *Sensors*, vol. 20, no. 5, 2020.

[238] T. H. A. Kolobe and A. H. Fagg, "Robot reinforcement and error-based movement learning in infants with and without cerebral palsy," *Physical Therapy*, vol. 99, no. 6, pp. 677–688, 2019.

[239] R. Ali, B. Kim, S. W. Kim, H. S. Kim, and F. Ishmanov, "(ReLBT): a Reinforcement learning-enabled listen before talk mechanism for LTE-LAA and Wi-Fi coexistence in IoT," *Computer Communications*, vol. 150, pp. 498–505, 2020.

[240] T. Yu and S. P. Zhang, "Automatic control of electricity generation based on 5-component update learning algorithm SARSA (λ)," *Kongzhi Lilun Yu Yingyong/Control Theory and Applications*, vol. 30, no. 10, pp. 1246–1251, 2013.

[241] C. Liu, J. Gao, Y. Bi, X. Shi, and D. Tian, "A multitasking-oriented robot arm motion planning scheme based on deep reinforcement learning and twin synchro-control," *Sensors*, vol. 20, no. 12, pp. 1–35, 2020.

[242] L. An and G.-H. Yang, "Opacity enforcement for confidential robust control in linear cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1234–1241, 2020.

[243] G. Faraci, C. Grasso, and G. Schembra, "Fog in the clouds: UAVs to provide edge computing to IoT devices," *ACM Transactions on Internet Technology*, vol. 20, no. 3, 2020.

[244] H. Joo, S. H. Ahmed, and Y. Lim, "Traffic signal control for smart cities using reinforcement learning," *Computer Communications*, vol. 154, pp. 324–330, 2020.

[245] Q. Wu, J. Wu, J. Shen, B. Yong, and Q. Zhou, "An edge based multi-agent auto communication method for traffic light control," *Sensors*, vol. 20, no. 15, pp. 1–16, 2020.

[246] T. Yu, S. Zhang, and Y. Hong, "Dynamic optimal CPS control for interconnected power systems based on SARSA Algorithm," *Lecture Notes in Electrical Engineering*, vol. 238, pp. 269–276, 2014.

[247] H. Xu, X. Liu, W. Yu, D. Griffith, and N. Golmie, "Reinforcement learning-based control and networking Co-design for industrial internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 885–898, 2020.

[248] T. Wang, X. Shen, M. S. Obaidat, X. Liu, and S. Wan, "Edge-learning-based hierarchical prefetching for collaborative information streaming in social IoT systems," *IEEE Transactions on Computational Social Systems*, p. 7, 2020.

[249] S. Liu, S. Li, and B. Xu, "Event-triggered resilient control for cyber-physical system under denial-of-service attacks," *International Journal of Control*, vol. 93, no. 8, pp. 1907–1919, 2020.

[250] S. Souihi, M. Souidi, and A. Mellouk, "An adaptive QoE-based network interface selection for multi-homed eHealth devices," *Internet of Things. IoT Infrastructures*, vol. 169, pp. 437–442, 2016.

[251] H. Van Dong, B. Quoc Khanh, N. Tran Lich, and N. T. Ngoc Anh, "Integrating multi-agent system, geographic information system, and reinforcement learning to simulate and optimize traffic signal control," *Recent Advances in Information and Communication Technology 2018*, vol. 769, pp. 145–154, 2019.

[252] V. Hakami, S. Mostafavi, N. T. Javan, and Z. Rashidi, "An optimal policy for joint compression and transmission control in delay-constrained energy harvesting IoT devices," *Computer Communications*, vol. 160, pp. 554–566, 2020.

[253] S. Kim, "One-player game based influential maximization scheme for social cloud service networks," *EAI/Springer Innovations in Communication and Computing*, pp. 175–184, Springer, Cham, Switzerland, 2019.

[254] D. Pacheco-Paramo and L. Tello-Oquendo, "Delay-aware dynamic access control for mMTC in wireless networks using deep reinforcement learning," *Computer Networks*, p. 182, 2020.

[255] Y. Zhao, J. Hu, K. Yang, and S. Cui, "Deep reinforcement learning aided intelligent access control in energy harvesting based WLAN," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14078–14082, 2020.

[256] Y. Hadjadj-Aoul and S. Ait-Chellouche, "Access control in nb-iot networks: a deep reinforcement learning strategy," *Information*, vol. 11, no. 11, pp. 1–16, 2020.

[257] S. Khairy, P. Balaprakash, L. X. Cai, and Y. Cheng, "Constrained deep reinforcement learning for energy sustainable multi-UAV based random access IoT networks with NOMA," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1101–1115, 2021.

[258] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," *IEEE Access*, vol. 8, pp. 209320–209344, 2020.

[259] F. Zhang, Q. Yang, and D. An, "CDDPG: a deep-reinforcement-learning-based approach for electric vehicle charging control," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3075–3087, 2021.

[260] W. Wu, F. Zhu, Y. Fu, and Q. Liu, "Deep deterministic policy gradient with clustered prioritized sampling," *Neural Information Processing*, vol. 11302, pp. 645–654, 2018.

[261] C. Cho, S. Shin, H. Jeon, and S. Yoon, "QoS-aware workload distribution in hierarchical edge clouds: a reinforcement learning approach," *IEEE Access*, vol. 8, pp. 193297–193313, 2020.

[262] A. Musaddiq, Z. Nain, Y. Ahmad Qadri, R. Ali, and S. W. Kim, "Reinforcement learning-enabled cross-layer optimization for low-power and lossy networks under heterogeneous traffic patterns," *Sensors*, vol. 20, no. 15, pp. 1–25, 2020.

[263] R. Huang, V. W. S. Wong, and R. Schober, "Throughput optimization for grant-free multiple access with multiagent deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 228–242, 2021.

[264] T. Yu and B. Zhou, "Reinforcement learning based CPS self-tuning control methodology for interconnected power systems," *Dianli Xitong Baohu yu Kongzhi/Power System Protection and Control*, vol. 37, no. 10, pp. 33–38, 2009.

[265] T. Yu and Y. Yuan, "An average reward model based whole process R (λ)-learning for optimal CPS control," *Dianli Xitong Zidonghua/Automation of Electric Power Systems*, vol. 34, no. 21, pp. 27–33, 2010.

[266] T. Liu, B. Tian, Y. Ai, and F.-Y. Wang, "Parallel reinforcement learning-based energy efficiency improvement

for a cyber-physical system," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 617–626, 2020.

[267] M. Kiermeier, S. Feld, T. Phan, and C. Linnhoff-Popien, "Anomaly detection in spatial layer models of autonomous agents," *Intelligent Data Engineering and Automated Learning - IDEAL 2018*, vol. 11314, pp. 156–163, 2018.

[268] S. Jeong, G. Yoo, M. Yoo, I. Yeom, and H. Woo, "Resource-efficient sensor data management for autonomous systems using deep reinforcement learning," *Sensors*, vol. 19, no. 20, 2019.

[269] M. Kadohisa, J. V. Verhagen, and E. T. Rolls, "The primate amygdala: neuronal representations of the viscosity, fat texture, temperature, grittiness and taste of foods," *Neuroscience*, vol. 132, no. 1, pp. 33–48, 2005.

[270] C. Schwenck, A. Ciaramidaro, M. Selivanova, J. Tournay, C. M. Freitag, and M. Siniatchkin, "Neural correlates of affective empathy and reinforcement learning in boys with conduct problems: fMRI evidence from a gambling task," *Behavioural Brain Research*, vol. 320, pp. 75–84, 2017.

[271] M. Li, W.-j. Liu, B. Lu, Y.-h. Wang, and J.-g. Liu, "Differential expression of Arc in the mesocorticolimbic system is involved in drug and natural rewarding behavior in rats," *Acta Pharmacologica Sinica*, vol. 34, no. 8, pp. 1013–1024, 2013.

[272] C. Wang, J. Wang, J. Wang, and X. Zhang, "Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6180–6190, 2020.

[273] H.-Y. Kim and J. Kim, "A load balancing scheme for gaming server applying reinforcement learning in IOT," *Computer Science and Information Systems*, vol. 17, no. 3, pp. 891–906, 2020.

[274] A. H. Keyhanipour, B. Moshiri, M. Rahgozar, F. Oroumchian, and A. A. Ansari, "Integration of data fusion and reinforcement learning techniques for the rank-aggregation problem," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1131–1145, 2016.

[275] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930–1941, 2019.

[276] M. Min, X. Wan, L. Xiao et al., "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, 2019.

[277] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered Public Safety IoT: a UAV-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.

[278] Y. Cui, D. Zhang, T. Zhang, L. Chen, M. Piao, and H. Zhu, "Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices," *AEU - International Journal of Electronics and Communications*, p. 118, 2020.

[279] K. Gai, K. Xu, Z. Lu, M. Qiu, and L. Zhu, "Fusion of cognitive wireless networks and edge computing," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 69–75, 2019.

[280] S. Spanò, G. C. Cardarilli, L. Di Nunzio et al., "An efficient hardware implementation of reinforcement learning: the q-learning algorithm," *IEEE Access*, vol. 7, pp. 186340–186351, 2019.

[281] M. S. Munir, S. F. Abedin, N. H. Tran, and C. S. Hong, "When edge computing meets microgrid: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7360–7374, 2019.

[282] N. Shoeibi and N. Shoeibi, "Future of smart parking: automated valet parking using deep Q-learning," *Advances in Intelligent Systems and Computing*, vol. 1004, pp. 177–182, 2020.

[283] Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "LiPSG: lightweight privacy-preserving Q-learning-based energy management for the IoT-enabled smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3935–3947, 2020.

[284] M. Ozturk, M. Jaber, and M. A. Imran, "Energy-aware smart connectivity for IoT networks: enabling smart ports," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[285] H. Lu, X. He, M. Du, X. Ruan, Y. Sun, and K. Wang, "Edge QoE: computation offloading with deep reinforcement learning for internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9255–9265, 2020.

[286] D. Ma, G. Lan, M. Hassan, W. Hu, and S. K. Das, "Sensing, computing, and communications for energy harvesting IoTs: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1222–1250, 2020.

[287] R. Bonnefoi, C. Moy, and J. Palicot, "Improvement of the LPWAN AMI backhaul's latency thanks to reinforcement learning algorithms," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–18, 2018.

[288] X. Bao, H. Liang, and L. Han, "Transmission optimization of social and physical sensor nodes via collaborative beamforming in cyber-physical-social systems," *Sensors*, vol. 18, no. 12, 2018.

[289] M. Han, J. Duan, S. Khairy, and L. X. Cai, "Enabling sustainable underwater IoT networks with energy harvesting: a decentralized reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9953–9964, 2020.

[290] T. Mohammed, A. Albeshri, I. Katib, and R. Mehmood, "UbiPriSEQ—deep reinforcement learning to manage privacy, security, energy, and QoS in 5G IoT hetnets," *Applied Sciences*, vol. 10, no. 20, pp. 1–18, 2020.

[291] J. Tang, H. Tang, X. Zhang et al., "Energy minimization in d2d-assisted cache-enabled internet of things: a deep reinforcement learning approach," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5412–5423, 2020.

[292] G. Maselli, M. Piva, and J. A. Stankovic, "Adaptive communication for battery-free devices in smart homes," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6977–6988, 2019.

[293] H. Ke, J. Wang, H. Wang, and Y. Ge, "Joint optimization of data offloading and resource allocation with renewable energy aware for IoT devices: a deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 179349–179363, 2019.

[294] Y. Xie, Z. Xu, J. Xu, S. Gong, and Y. Wang, "Backscatter-Aided hybrid data offloading for mobile edge computing via deep reinforcement learning," *LNICST*, vol. 294, 2019.

[295] Y. Rioual, J. Laurent, and J.-P. Diguet, "Reinforcement-learning approach guidelines for energy management," *Journal of Low Power Electronics*, vol. 15, no. 3, pp. 283–293, 2019.

[296] C. Han, A. Liu, H. Wang, L. Huo, and X. Liang, "Dynamic anti-jamming coalition for satellite-enabled army IoT: a distributed game approach," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10932–10944, 2020.

[297] F. Jiang, K. Wang, L. Dong, C. Pan, and K. Yang, "Stacked autoencoder-based deep reinforcement learning for online resource scheduling in large-scale MEC networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9278–9290, 2020.

[298] X. Tu, C. Xu, S. Liu et al., "Efficient monocular depth estimation for edge devices in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2821–2832, 2021.

[299] J. Long, Y. Luo, X. Zhu, E. Luo, and M. Huang, "Computation offloading through mobile vehicles in IoT-edge-cloud network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.

[300] Y. Akbari and S. Tabatabaei, "A new method to find a high reliable route in IoT by using reinforcement learning and fuzzy logic," *Wireless Personal Communications*, vol. 112, no. 2, pp. 967–983, 2020.

[301] Y. Li, X. Zhao, and H. Liang, "Throughput maximization by deep reinforcement learning with energy cooperation for renewable ultradense IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9091–9102, 2020.

[302] J. Zheng, L. Gao, H. Wang et al., "Smart edge caching-aided partial opportunistic interference alignment in HetNets," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1842–1850, 2020.

[303] M. Peng, S. Garg, X. Wang, A. Bradai, H. Lin, and M. S. Hossain, "Learning-based IoT data aggregation for disaster scenarios," *IEEE Access*, vol. 8, pp. 128490–128497, 2020.

[304] S. Sarwar, R. Sirhindi, L. Aslam, G. Mustafa, M. M. Yousaf, and S. W. U. Q. Jaffry, "Reinforcement learning based adaptive duty cycling in LR-WPANs," *IEEE Access*, vol. 8, pp. 161157–161174, 2020.

[305] K. Wang, C. M. Chen, M. S. Hossain, G. Muhammad, S. Kumar, and S. Kumari, "Transfer reinforcement learning-based road object detection in next generation IoT domain," *Computer Networks*, p. 193, 2021.

[306] M. I. Khan, L. Reggiani, M. M. Alam et al., "Q-learning based joint energy-spectral efficiency optimization in multi-hop device-to-device communication," *Sensors*, vol. 20, no. 22, pp. 1–23, 2020.

[307] S. F. Abedin, M. S. Munir, N. H. Tran, Z. Han, and C. S. Hong, "Data freshness and energy-efficient UAV navigation optimization: a deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5994–6006, 2020.

[308] G. Kaur, P. Chanak, and M. Bhattacharya, "Energy efficient intelligent routing scheme for IoT-enabled WSNs," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11440–11449, 2021.

[309] Z. Xiong, Y. Zhang, W. Y. B. Lim et al., "UAV-assisted wireless energy and data transfer with deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 85–99, 2021.

[310] Y. Nie, J. Zhao, J. Liu, J. Jiang, and R. Ding, "Energy-efficient UAV trajectory design for backscatter communication: a deep reinforcement learning approach," *China Communications*, vol. 17, no. 10, pp. 129–141, 2020.

[311] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[312] F. M. Al-Turjman, "Information-centric sensor networks for cognitive IoT: an overview," *Annales des Telecommunications/Annals of Telecommunications*, vol. 72, no. 1-2, pp. 3–18, 2017.

[313] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11158–11168, 2019.

[314] H. Khelifi, S. Luo, B. Nour et al., "Bringing deep learning at the edge of information-centric internet of things," *IEEE Communications Letters*, vol. 23, no. 1, pp. 52–55, 2019.

[315] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: a comprehensive survey," *Ad Hoc Networks*, vol. 93, 2019.

[316] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Optimizing and updating lora communication parameters: a machine learning approach," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 884–895, 2019.

[317] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled internet of things: network architecture and spectrum access," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 44–51, 2020.

[318] A. Foerster, A. Udugama, C. Görg, K. Kuladinithi, A. Timm-Giel, and A. Cama-Pinto, "A novel data dissemination model for organic data flows," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 158, pp. 239–252, 2015.

[319] S. Shukla, M. F. Hassan, M. K. Khan, L. T. Jung, and A. Awang, "An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment," *PLoS ONE*, vol. 14, no. 11, Article ID e0224934, 2019.

[320] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1974–1993, 2020.

[321] R. Bajracharya, R. Shrestha, and S. W. Kim, "Q-learning based fair and efficient coexistence of LTE in unlicensed band," *Sensors*, vol. 19, no. 13, 2019.

[322] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6242–6251, 2020.

[323] Z. Cao, P. Zhou, R. Li, S. Huang, and D. Wu, "Multiagent deep reinforcement learning for joint multichannel access and task offloading of mobile-edge computing in industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6201–6213, 2020.

[324] M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep learning at the mobile edge: opportunities for 5G networks," *Applied Sciences*, vol. 10, no. 14, 2020.

[325] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[326] A. Serhani, N. Naja, and A. Jamali, "AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET-IoT systems," *Cluster Computing*, vol. 23, no. 1, pp. 13–27, 2020.

[327] F. Jameel, U. Javaid, W. U. Khan, M. N. Aman, H. Pervaiz, and R. Jäntti, "Reinforcement learning in blockchain-enabled IIoT networks: a survey of recent advances and open challenges," *Sustainability*, vol. 12, no. 12, 2020.

[328] J. M. C. Neto, S. F. G. Neto, P. M. de Santana, and V. A. de Sousa, "Multi-cell LTE-U/Wi-Fi coexistence evaluation using a reinforcement learning framework," *Sensors*, vol. 20, no. 7, 2020.

[329] W. Ejaz, M. Basharat, S. Saadat, A. M. Khattak, M. Naeem, and A. Anpalagan, "Learning paradigms for communication and computing technologies in IoT systems," *Computer Communications*, vol. 153, pp. 11–25, 2020.

[330] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K.-K. Wong, "Minimum throughput maximization for multi-UAV enabled WPCN: a deep reinforcement learning method," *IEEE Access*, vol. 8, pp. 9124–9132, 2020.

[331] A. Abane, M. Daoui, S. Bouzefrane, and P. Muhlethaler, "A lightweight forwarding strategy for named data networking in low-end IoT," *Journal of Network and Computer Applications*, p. 148, 2019.

[332] R. Ali, Y. B. Zikria, B.-S. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for dense wireless networks in smart cities," *EAI/Springer Innovations in Communication and Computing, Springer, Cham, Switzerland*, pp. 43–70, 2020.

[333] Q. Zhang, Y.-C. Liang, and H. V. Poor, "Intelligent user association for symbiotic radio networks using deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4535–4548, 2020.

[334] R. M. Sandoval, S. Canovas-Carrasco, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "A reinforcement learning-based framework for the exploitation of multiple rats in the iot," *IEEE Access*, vol. 7, pp. 123341–123354, 2019.

[335] C. Sun, H. Ding, and X. Liu, "Multichannel spectrum access based on reinforcement learning in cognitive internet of things," *Ad Hoc Networks*, vol. 106, 2020.

[336] G. L. Santos, P. T. Endo, D. Sadok, and J. Kelner, "When 5G meets deep learning: a systematic review," *Algorithms*, vol. 13, no. 9, 2020.

[337] N. Chen, T. Qiu, C. Mu, M. Han, and P. Zhou, "Deep actor-critic learning-based robustness enhancement of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6191–6200, 2020.

[338] Y. Zhang, B. Feng, W. Quan et al., "Cooperative edge caching: a multi-agent deep learning based approach," *IEEE Access*, vol. 8, pp. 133212–133224, 2020.

[339] Y. Hao, M. Li, D. Wu, M. Chen, M. M. Hassan, and G. Fortino, "Human-like hybrid caching in software-defined edge cloud," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5806–5815, 2020.

[340] F. Dou, J. Lu, T. Xu, C.-H. Huang, and J. Bi, "A bisection reinforcement learning approach to 3-D indoor localization," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6519–6535, 2021.

[341] T.-W. Ban, "An autonomous transmission scheme using dueling DQN for D2D communication networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16348–16352, 2020.

[342] K. S. Shin, G. H. Hwang, and O. Jo, "Distributed reinforcement learning scheme for environmentally adaptive IoT network selection," *Electronics Letters*, vol. 56, no. 9, pp. 441–444, 2020.

[343] N. Garg, M. Sellathurai, V. Bhatia, and T. Ratnarajah, "Function approximation based reinforcement learning for edge caching in massive MIMO networks," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2304–2316, 2021.

[344] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.

[345] T. S. P. Kumar and P. V. Krishna, "Power modelling of sensors for IoT using reinforcement learning," *International Journal of Advanced Intelligence Paradigms*, vol. 10, no. 1-2, pp. 3–22, 2018.

[346] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, "Deep reinforcement learning (DRL): another perspective for unsupervised wireless localization," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6279–6287, 2020.

[347] M. Li, C. Chen, C. Hua, and X. Guan, "Intelligent latency-aware virtual network embedding for industrial wireless networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7484–7496, 2019.

[348] G. M. Dias, C. B. Margi, F. C. P. De Oliveira, and B. Bellalta, "Cloud-empowered, self-managing wireless sensor networks: interconnecting management operations at the application layer," *IEEE Consumer Electronics Magazine*, vol. 8, no. 1, pp. 55–60, 2019.

[349] C. Yang, K.-W. Chin, T. He, and Y. Liu, "On sampling time maximization in wireless powered internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 641–650, 2019.

[350] A. Kawewong, Y. Honda, M. Tsuboyama, and O. Hasegawa, "A common-neural-pattern based reasoning for mobile robot cognitive mapping," *Advances in Neuro-Information Processing*, vol. 5506, pp. 32–39, 2009.

[351] C. Moy, L. Besson, G. Delbarre, and L. Toutain, "Decentralized spectrum learning for radio collision mitigation in ultra-dense IoT networks: LoRaWAN case study and experiments," *Annales des Telecommunications/Annals of Telecommunications*, vol. 75, no. 11-12, pp. 711–727, 2020.

[352] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for internet of things: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2074–2083, 2019.

[353] Z. Li, Y. Lu, Y. Shi, Z. Wang, W. Qiao, and Y. Liu, "A Dyna-Q-based solution for UAV networks against smart jamming attacks," *Symmetry*, vol. 11, no. 5, 2019.

[354] J. Luo, S. Green, P. Feghali, G. Legrady, and C. K. Koç, *Reinforcement Learning and Trustworthy Autonomy*, Springer International Publishing, New York, NY, USA, 2018.

[355] Y. Liu, W. Zhang, S. Pan, Y. Li, and Y. Chen, "Analyzing the robotic behavior in a smart city with deep enforcement and imitation learning using IoRT," *Computer Communications*, vol. 150, pp. 346–356, 2020.

[356] K. Watanabe and S. Inada, "Search algorithm of the assembly sequence of products by using past learning results," *International Journal of Production Economics*, p. 226, 2020.

[357] K. Baek and I.-Y. Ko, "Effect-driven selection of web of things services in cyber-physical systems using reinforcement learning," *Lecture Notes in Computer Science*, vol. 11496, pp. 554–559, 2019.

[358] I. Verner, M. Reitman, D. Cuperman, T. Yan, E. Finkelstein, and T. Romm, "Exposing robot learning to students in augmented reality experience," *Smart Industry & Smart Education*, vol. 47, pp. 610–619, 2019.

[359] Y.-T. Tsai, C.-H. Lee, T.-Y. Liu et al., "Utilization of a reinforcement learning algorithm for the accurate alignment of a robotic arm in a complete soft fabric shoe tongues automation process," *Journal of Manufacturing Systems*, vol. 56, pp. 501–513, 2020.

[360] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *International Journal of*

Advanced Manufacturing Technology, vol. 109, no. 3-4, pp. 953–974, 2020.

[361] D. B. Noureddine, M. Krichen, S. Mechti, T. Nahhal, and W. Y. H. Adoni, "An agent-based architecture using deep reinforcement learning for the intelligent internet of things applications," *Advances on Smart and Soft Computing*, vol. 1188, pp. 273–283, 2021.

[362] R. S. Alonso, "Deep tech and artificial intelligence for worker safety in robotic manufacturing environments," *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*, vol. 1242, pp. 234–240, 2021.

[363] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624–635, 2018.

[364] X. Zhang, L. Yao, S. Zhang, S. Kanhere, M. Sheng, and Y. Liu, "Internet of things meets brain-computer interface: a unified deep learning framework for enabling human-thing cognitive interactivity," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2084–2092, 2019.

[365] O. A. Sianaki, A. Yousefi, A. R. Tabesh, and M. Mahdavi, "Machine learning applications: the past and current research trend in diverse industries," *Inventions*, vol. 4, no. 1, 2019.

[366] G. Neelakantam, D. D. Onthoni, and P. K. Sahoo, "Reinforcement learning based passengers assistance system for crowded public transportation in fog enabled smart city," *Electronics (Switzerland)*, vol. 9, no. 9, pp. 1–19, 2020.

[367] M. Rivas and F. Giorno, "A reinforcement learning multi-agent architecture prototype for smart homes (IoT)," *Proceedings of the Future Technologies Conference (FTC) 2018*, vol. 880, pp. 159–170, 2019.

[368] N. Magaia, R. Fonseca, K. Muhammad, A. H. F. N. Segundo, A. V. Lira Neto, and V. H. C. De Albuquerque, "Industrial internet-of-things security enhanced with deep learning approaches for smart cities," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6393–6405, 2021.

[369] S. Pan, P. Li, D. Zeng, S. Guo, and G. Hu, "A ${Q}$ -learning based framework for congested link identification," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9668–9678, 2019.

[370] H. Lin, Z. Chen, and L. Wang, "Offloading for edge computing in low power wide area networks with energy harvesting," *IEEE Access*, vol. 7, pp. 78919–78929, 2019.

[371] D. K. Sharma, J. J. P. C. Rodrigues, V. Vashishth, A. Khanna, and A. Chhabra, "RLProph: a dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks," *Wireless Networks*, vol. 26, no. 6, pp. 4319–4338, 2020.

[372] B. Song, J. Song, and J. Ye, "A dynamic pricing mechanism in IoT for DaaS: a reinforcement learning approach," *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, vol. 1075, pp. 604–615, 2020.

[373] D. Wang, X. Tian, H. Cui, and Z. Liu, "Reinforcement learning-based joint task offloading and migration schemes optimization in mobility-aware MEC network," *China Communications*, vol. 17, no. 8, pp. 31–44, 2020.

[374] W. Shafik, S. Mojtaba Matinkhah, P. Etemadinejad, and M. N. Sanda, "Reinforcement learning rebirth, techniques, challenges, and resolutions," *International Journal on Informatics Visualization*, vol. 4, no. 3, pp. 127–135, 2020.

[375] E. Erdemir, P. L. Dragotti, and D. Gunduz, "Privacy-aware time-series data sharing with deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 389–401, 2021.

[376] P. Wang, L. T. Yang, J. Li, X. Li, and X. Zhou, "MMDP: a mobile-IoT based multi-modal reinforcement learning service framework," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 675–684, 2020.

[377] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-agent reinforcement learning based cooperative content caching for mobile edge networks," *IEEE Access*, vol. 7, pp. 61856–61867, 2019.

[378] J. Ma, S. Hasegawa, S. J. Kim, and M. Hasegawa, "A reinforcement-learning-based distributed resource selection algorithm for massive IoT," *Applied Sciences*, vol. 9, no. 18, 2019.

[379] Y. Qian, L. Shi, J. Li et al., "A workflow-aided internet of things paradigm with intelligent edge computing," *IEEE Network*, vol. 34, no. 6, pp. 92–99, 2020.

[380] Z. Shi, Y. Zeng, and Z. Wu, "Service chain orchestration based on deep reinforcement learning in intent-based IoT," *Proceedings of the 9th International Conference on Computer Engineering and Networks*, vol. 1143, pp. 875–882, 2021.

[381] W.-C. Chien, H.-Y. Weng, and C.-F. Lai, "Q-learning based collaborative cache allocation in mobile edge computing," *Future Generation Computer Systems*, vol. 102, pp. 603–610, 2020.

[382] V. Vijayaraghavan and J. R. Leevinson, *Intelligent Traffic Management Systems for Next Generation IoV in Smart City Scenario*, Springer International Publishing, New York, NY, USA, 2020.

[383] T. Lee, O. Jo, and K. Shin, "CoRL: collaborative reinforcement learning-based MAC protocol for IoT networks," *Electronics (Switzerland)*, vol. 9, no. 1, 2020.

[384] C. Kim, "Deep reinforcement learning by balancing offline Monte Carlo and online temporal difference use based on environment experiences," *Symmetry*, vol. 12, no. 10, pp. 1–16, 2020.

[385] S. Misra, P. K. Deb, N. Koppala, A. Mukherjee, and S. Mao, "S-nav: safety-aware IoT navigation tool for avoiding COVID-19 hotspots," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6975–6982, 2021.

[386] D. N. Doan, D. Zaharie, and D. Petcu, "Auto-scaling for a streaming architecture with fuzzy deep reinforcement learning," *Lecture Notes in Computer Science*, vol. 11997, 2020.

[387] Y. Liu, H. Wang, M. Peng, J. Guan, and Y. Wang, "An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8616–8631, 2021.

[388] H. Guo, S. Li, B. Li, Y. Ma, and X. Ren, "A new learning automata-based pruning method to train deep neural networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3263–3269, 2018.

[389] Y. Wang, X. Chen, L. Wang, and G. Min, "Effective IoT-facilitated storm surge flood modeling based on deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6338–6347, 2020.

[390] J. Yun, Y. Goh, and J.-M. Chung, "DQN-based optimization framework for secure sharded blockchain systems," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 708–722, 2021.

[391] K. E. Mwangi, S. Masupe, and J. Mandu, "Modelling malware propagation on the internet of things using an agent-based approach on complex networks," *Jordanian Journal of Computers and Information Technology*, vol. 6, no. 1, pp. 26–40, 2020.

[392] X. He, K. Wang, and W. Xu, "QoE-Driven content-centric caching with deep reinforcement learning in edge-enabled

IoT," *IEEE Computational Intelligence Magazine*, vol. 14, no. 4, pp. 12–20, 2019.

[393] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, and C. So-In, "Efficient SDN-based traffic monitoring in IoT networks with double deep Q-network," *Lecture Notes in Computer Science*, vol. 12575, 2020 LNCS:26–38.

[394] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[395] B. Banerjee and L. Kraemer, "Action discovery for single and multi-agent reinforcement learning," *Advances in Complex Systems*, vol. 14, no. 2, pp. 279–305, 2011.

[396] Y. Li, F. Qi, Z. Wang, X. Yu, and S. Shao, "Distributed edge computing offloading algorithm based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 85204–85215, 2020.

[397] L. Zhou, Q. Liu, F. Wu, and Y. Wei, *Deep Learning Based Sensing Resource Allocation for Mobile Target Tracking*, pp. 430–435, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ, USA, 2020.

[398] P. Zhang, Y. Yuan, Z. Wang, and C. Sun, *A Hierarchical Game Approach to the Coupled Resilient Control of CPS against Denial-Of-Service Attack*, pp. 15–20, IEEE Computer Society, Washington, DC, USA, 2019.