Deep Reinforcement Learning for Internet of Things: A Comprehensive Survey

Wuhui Chen, *Member, IEEE*, Xiaoyu Qiu, Ting Cai, Hong-Ning Dai, *Senior Member, IEEE*, Zibin Zheng, *Senior Member, IEEE*, and Yan Zhang, *Fellow, IEEE*

***Abstract*—The incumbent Internet of Things suffers from poor scalability and elasticity exhibiting in communication, computing, caching and control (4Cs) problems. The recent advances in deep reinforcement learning (DRL) algorithms can potentially address the above problems of IoT systems. In this context, this paper provides a comprehensive survey that overviews DRL algorithms and discusses DRL-enabled IoT applications. In particular, we first briefly review the state-of-the-art DRL algorithms and present a comprehensive analysis on their advantages and challenges. We then discuss on applying DRL algorithms to a wide variety of IoT applications including smart grid, intelligent transportation systems, industrial IoT applications, mobile crowdsensing, and blockchain-empowered IoT. Meanwhile, the discussion of each IoT application domain is accompanied by an in-depth summary and comparison of DRL algorithms. Moreover, we highlight emerging challenges and outline future research directions in driving the further success of DRL in IoT applications.**

***Index Terms*—Deep reinforcement learning, Internet of Things, decision making, resource allocation.**

I. Introduction

In RECENT years, Internet of Things (IoT) has appeared as a paradigm to drive the evolution of modern industries and smart cities. IoT essentially consists of "things" including hand-held devices, healthcare devices, various tags, actuators and sensors, which are connected with the Internet via IoT gateways, access points and base stations [1]. IoT exhibits the following key features: 1) *large scale*: the number of IoT devices is expected to reach 24 billion in 2020 [2]; 2) *heterogeneous* IoT devices and IoT data: various IoT devices can generate structured, semi-structured and unstructured IoT data; 3) *complex* IoT networks: IoT composites of a great diversity of IoT networks enabled by different types of IoT protocols such as 6LoWPAN, LPWAN, WLAN, 4G and 5G networks; 4) *resource constraints*: IoT is suffering from intrinsic resource constraints such as poor computing capability, limited storage and low battery capacity of IoT devices; 5) *security and privacy vulnerabilities*: owning to the decentralization of IoT systems and resource constraints of IoT devices, IoT systems have a number of security and privacy vulnerabilities.

The advent of cloud computing and edge computing brings the opportunities to overcome the underlying limitations of IoT systems. In particular, remote cloud servers can provide IoT devices with abundant computing and storage resources so as to extend the capabilities of IoT. However, accompanied by the benefits, cloud computing also brings a number of challenges to IoT systems, including long latency, privacy-leakage risk, failure to support context-aware applications. Fortunately, the recent advances in multi-access edge computing (MEC) may tackle the above issues of cloud computing. Specifically, MEC can offload the computing tasks from remote cloud servers to MEC servers, which are deployed close to IoT devices so as to reduce the latency, improve the context-awareness and protect data privacy. Thus, the orchestration of cloud computing and MEC can further enhance IoT systems, whereas a number of research issues need to be addressed.

The main obstacles to fully enable the collaboration of cloud computing, MEC and IoT exhibit in Communication, Computing, Caching and Control (4Cs) problems. In particular, both the heterogeneity of IoT systems and the complexity of IoT networks bring the challenges in communications. For example, it is challenging to manage a network to connect different types of IoT devices to fulfill various user demands. Moreover, it is also difficult to decompose a computing task into a number of sub-tasks and schedule them to be executed at either edge server or remote cloud server in an elastic manner. In addition, IoT data is distributed across IoT devices, MEC servers or cloud servers. The appropriate storage (or caching) of IoT data at different locations according to the "interests" is a necessity to satisfy context-aware and latency-sensitive applications, whereas it is quite difficult to achieve this goal own to the challenge in predicting the varied interests in advance. Furthermore, it is also challenging to control

Manuscript received October 7, 2020; revised February 27, 2021; accepted April 5, 2021. Date of publication April 13, 2021; date of current version August 23, 2021. This work was supported in part by the National Key Research and Development Plan under Grant 2018YFB1003800; in part by the Macao Science and Technology Development Fund through Macao Funding Scheme for Key Research and Development Projects under Grant 0025/2019/AKP; in part by the National Natural Science Foundation of China under Grant 61802450; in part by the Natural Science Foundation of Guangdong under Grant 2018A030313005; in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X355; and in part by the Pearl River Talent Recruitment Program under Grant 2019QN01X130. *(Corresponding authors: Zibin Zheng; Hong-Ning Dai.)*

Wuhui Chen, Xiaoyu Qiu, Ting Cai, and Zibin Zheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China, and also with the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenwuh@mail.sysu.edu.cn; qiuxy23@mail2.sysu.edu.cn; cait9@mail2.sysu.edu.cn; zhzibin@mail.sysu.edu.cn).

Hong-Ning Dai is with the Faculty of Information Technology, Macau University of Science and Technology, Macau, China (e-mail: hndai@ieee.org).

Yan Zhang is with the Department of Informatics, University of Oslo, 0373 Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, 1325 Lysaker, Norway (e-mail: yanzhang@ieee.org).

Digital Object Identifier 10.1109/COMST.2021.3073036

IoT systems to accommodate the environmental dynamics and make intellectual decisions or actions in a real time manner.

The 4Cs problems can be formulated as cooperative or non-cooperative games, Markov Decision Processes (MDPs), combinatorial optimization problems. Since most of these problems are NP-hard, genetic algorithms, multi-stage heuristic algorithms, simulated annealing algorithm, particle swarm algorithm, Lyapunov algorithm and Lagrangian relaxation approaches have been widely adopted to solve them through reducing the computational complexity especially for large-scale IoT networks [3]. However, the above algorithms cannot be suitable for real-time decision making because the 4Cs optimization problems often require a considerable number of iterations to find a satisfactory solution. To make the situation even worse, it often requires the 4Cs problems to be solved to adapt to the time-varying environments. In addition, as most of traditional approaches only consider one-shot optimization, it is challenging for them to adjust the policy to accommodate varied environments so as to achieve a stable long-term performance.

A. Advances in Deep Reinforcement Learning

Fortunately, the advances in deep reinforcement learning (DRL) have shown great potentials to solve the above 4Cs problems [4]. DRL algorithms essentially leverage powerful function approximation properties of deep neural networks (DNNs) to efficiently remove the curse of high dimensionality and complexity of problems. Meanwhile, DRL schemes also learn from both online and offline training samples, and consequently establish an optimal correlation between each state-action pair and its associated value (i.e., cumulative reward) [5]. A number of recent studies on using DRL in IoT systems have shown the following advantages.

- *Long term performance optimization:* Considering the high-complexity and uncertainty of environments, most of the traditional approaches focus only on one-shot optimization, which may fail to achieve the stable long term optimized performance. In contrast, the experience-driven DRL is able to learn the optimal decision-making policy based on historical experience.
- *Real-time decision making:* Although the trial-and-error training process of DRL is a time-consuming and resource-consuming process, once converged, DRL agents can respond to the environment changes in several milliseconds to achieve real-time decision making.
- *Online learning without prior knowledge:* Comparing with traditional heuristic algorithms, the experience-driven DRL makes no assumption on the environment models. In particular, DRL can improve its policy during operations by accumulating new experiences, thereby being easily adapted to environmental changes.
- *High scalability:* Conventional reinforcement learning (RL) algorithms cannot handle the explosion of state space with the increased complexity of IoT systems. Fortunately, DRL successfully exploits DNNs as function approximators to extend the scalability of RL to accommodate complex IoT systems.

TABLE I
COMPARISON OF THIS SURVEY WITH REPRESENTATIVE SURVEY PAPERS

Research issues	[9]	[10]	This survey
Communication	✓	×	✓
Computation	✓	✓	✓
Caching	✓	✓	✓
Control	✓	✓	✓
Domain-oriented applications	×	×	✓
Privacy, security and trust	×	×	✓

B. Prior Studies

There are several surveys on IoT technologies and DRL. For examples, Al-Fuqaha *et al.* [1] presented a survey of IoT with an emphasis on enabling protocols, technologies, and application issues while the studies of [6] and [7] presented the reviews on using deep learning techniques to IoT data analytics. Meanwhile, Arulkumaran *et al.* [8] presented an overview on DRL by introducing several typical DRL algorithms (such as deep *Q*-network, policy search, and actor-critic network) despite no IoT technologies being discussed. In summary, the above studies address either IoT or DRL issues only.

To the best of our knowledge, there are only two survey/overview articles focusing on using DRL on network and communications. In particular, Luong *et al.* [9] studied the applications of DRL in communications and networking whereas the focus of that paper mainly lies in communications and networking problems. Lei *et al.* [10] gave a survey on the usage of DRL for autonomous IoT. Although these efforts have provided valuable references and insights, there are still some emerging challenges that have not been considered in these studies. For examples, the emerging blockchain-empowered IoT applications may have strict trust and privacy requirements, which have not been well addressed in the current literature. This absence motivates us to deliver a comprehensive survey and provide a state-of-the-art literature review on a wide variety of IoT applications enabled by DRL algorithms. Most importantly, the above critical 4Cs problems as well as other crucial issues like security, privacy and trust are also expected to be addressed. Table I presents a comparison of this paper with the most representative surveys such as [9] and [10].

C. Contributions

In this article, we present a comprehensive and systematic review of the recent studies on using DRL algorithms to address the 4Cs problems so as to enable a wide variety of IoT applications. Fig. 1 shows the taxonomy of DRL-enabled IoT applications. This timely study makes the following contributions:

- *Summaries and classifications of DRL algorithms:* We review the state-of-the-art DRL algorithms. Meanwhile, we also discuss the internal mechanisms, advantages and existing challenges of DRL algorithms.
- *DRL applications in different domains:* We provide a comprehensive review of a wide diversity of

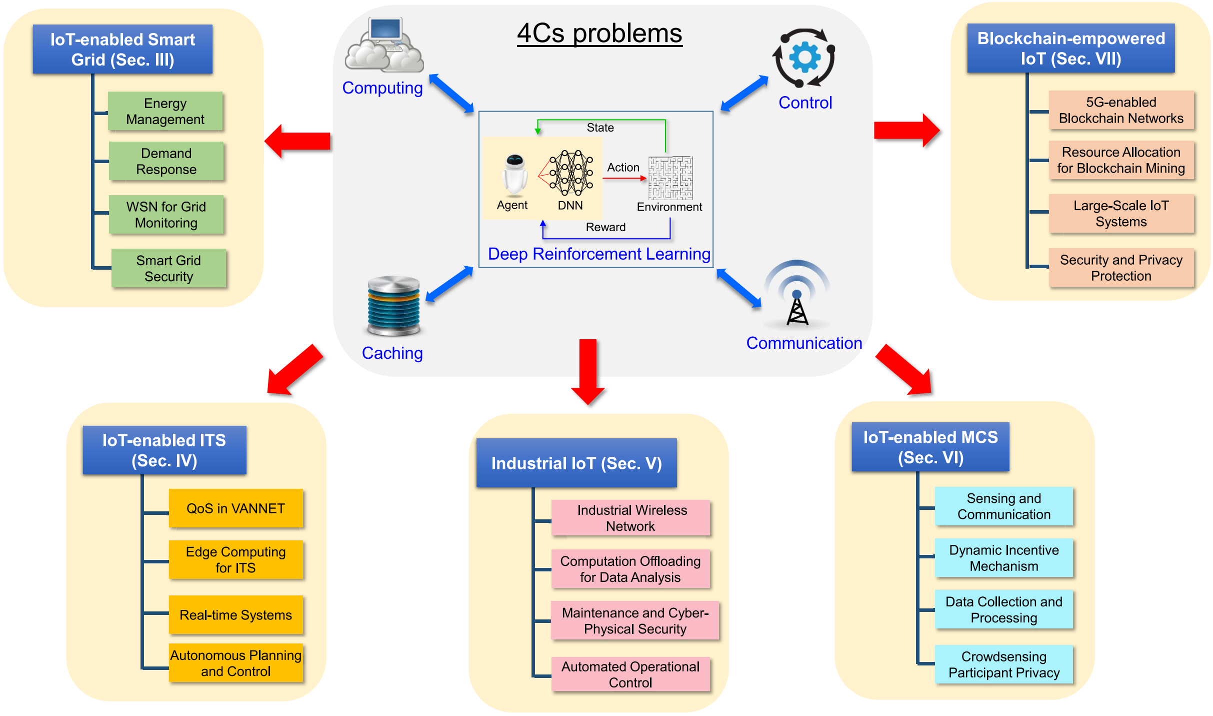

IoT-enabled Smart Grid (Sec. III)
Energy Management
Demand Response
WSN for Grid Monitoring
Smart Grid Security
4Cs problems
Computing
Control
State
Action
Agent
DNN
Environment
Reward
Deep Reinforcement Learning
Caching
Communication
Blockchain-empowered IoT (Sec. VII)
5G-enabled Blockchain Networks
Resource Allocation for Blockchain Mining
Large-Scale IoT Systems
Security and Privacy Protection
IoT-enabled ITS (Sec. IV)
QoS in VANNET
Edge Computing for ITS
Real-time Systems
Autonomous Planning and Control
Industrial IoT (Sec. V)
Industrial Wireless Network
Computation Offloading for Data Analysis
Maintenance and Cyber-Physical Security
Automated Operational Control
IoT-enabled MCS (Sec. VI)
Sensing and Communication
Dynamic Incentive Mechanism
Data Collection and Processing
Crowdsensing Participant Privacy

Fig. 1. Taxonomy of DRL applications in IoT.

DRL-enabled IoT applications including smart grid, intelligent transportation system (ITS), industrial IoT (IIoT) applications, mobile crowdsensing (MCS), blockchain-empowered IoT. With special attention on how to apply DRL to different applications, we also provide a comparison and guideline for using different DRL algorithms in various IoT applications.

- *Challenges and future trends:* We highlight emerging critical challenges and outline future research directions in driving the further success of DRL in IoT applications.

The remainder of the paper is structured as follows. In Section II, we first describe the 4Cs problems and present a review of DRL. Sections III, IV, V, VI, and VII then present DRL applications in IoT-enabled smart grid, IoT-enabled ITS, IIoT, IoT-enabled MCS and blockchain-empowered IoT, respectively. Section VIII next outlines challenges and future directions. Section IX finally concludes the paper. Table II gives the list of commonly used acronyms.

II. 4Cs Problems and DRL

A. 4Cs Problems

The potential of IoT has encouraged its integration in many fields including smart grid, ITS, industry, etc. The large-scale interconnection of IoT devices and the derivative 4Cs problems require stakeholders to have a clear understanding of their building blocks, unexploited potential, and existing challenges. Based on the characteristics of IoT, a plethora of work has been devoted to addressing the 4Cs problems from different aspects. We present the major research topics as follows:

1) Real-Time Data Collection and Processing: In the era of IoT, massive sensing devices will be deployed to collect/generate sensory data for various applications. IoT data analysis provides the opportunity to explore new information, gain worthy insights and make comprehensive decisions, making IoT a valuable paradigm. However, applying analysis over such enormous amounts of data along with meeting the real-time requirements is challenging. To address the corresponding 4Cs problems, more efforts are expected to shift the conventional approaches to emerging technologies such as big data and artificial intelligence.

Large-Scale IoT Connectivity and Networking: Connectivity and networking are the foundations of future IoT communication systems. However, the large-scale interconnection of IoT devices defies the very design of current communication networks. Traditional centralized, server/client paradigms are not sufficient to support the future IoT ecosystems that join billions of devices. In fact, future IoT ecosystems largely depend on the decentralization of IoT networks. It is reasonable to expect that emerging powerful communication protocols such as 5G will deliver enhancements. This demands for a more efficient, flexible, and agile wireless network system.

3) MEC Resource Allocation: Typically, the resource allocation of MEC is a highly sophisticated integrated optimization problem of computing, caching and communication, making it difficult to solve with traditional methods. So far, there is no common agreed upon framework for the MEC of IoT. Therefore, it is important to design a resource allocation strategy that considers multiple factors, such as system scalability, the heterogeneity and interoperability of IoT devices, and the trading between resource suppliers and customers.

4) IoT Scheduling and Maintenance: The widespread deployment of IoT devices and the emergence of various IoT

TABLE II
LIST OF COMMONLY USED ACRONYMS

Acronyms	Descriptions
4C	Computing, Communication, Caching and Control
DNN	Deep Neural Network
DL	Deep Learning
DPG/DDPG	Deterministic Policy Gradient/ Deep Deterministic Policy Gradient
DQN	Deep Q-Network
IIoT	Industrial IoT
IoT	Internet of Things
ITS	Intelligent Transportation System
LSTM	Long and Short Time Memory
MARL	Multi-Agent Reinforcement Learning
MAS	Multi-Agent System
MEC	Multi-access Edge Computing
MCS	Mobile Crowdsensing
MDP/POMDP	Markov Decision Process/ Partially Observable MDP
QoS	Quality-of-Service
RL/DRL	Reinforcement learning/ Deep Reinforcement Learning
RSU	Road Side Unit
SARSA	State-Action-Reward-State-Action
SDN/SDR	Software Defined Network/ Software Defined Radio
UAV	Unmanned Aerial Vehicle
VANET	Vehicular Ad-hoc Network

applications open up new profit sources such as consumer IoT, commercial IoT, and IIoT. However, this also presents control challenges. The accurate mathematical models for such application scenarios are difficult to obtain due to their complexity and dynamics. It is challenging to design an algorithm that can control IoT without the awareness of system models. In addition, the application of IoT also puts forward higher requirements for system stability. Therefore, a well-conceived preventive maintenance strategy is crucial to prevent the potential system breakdown.

5) Energy Efficiency Management: The energy efficiency management strategy has a direct bearing on the computing performance of IoT systems. On the one hand, IoT devices are typically constrained with limited computational power and energy supply. On the other hand, because IoT devices can be located anywhere (such as a harsh, remote environment), it is generally unrealistic to run wires to IoT devices. It is natural to envision the wide application of wireless charging technologies in future IoT ecosystems, in which the erratic nature of wireless energy harvesting is the major bottleneck.

6) Security and Privacy Concerns: Security and privacy issues are the major barriers in deploying IoT devices and achieving automatic control in real-world scenarios. Due to the decentralization and resource limitations of most IoT devices, there are many security vulnerabilities in IoT systems. Security technologies such as blockchain are developing, providing prospects for the realization of a secure and privacy-preserving IoT system. However, there is a lack of intelligent strategies to unleash the power of these technologies.

The 4Cs problems have been widespread concerned in recent years. Approaches such as heuristic algorithm, Lyapunov algorithm and game theory have been widely adopted in various IoT applications. However, the following issues are not given sufficient attention in traditional approaches:

1) Most of these approaches consider one-shot optimization, while the system dynamics play an important role in the optimization of IoT systems.
2) Even for the few approaches that consider long-term optimization, most of them are founded on assumptions about system models, which may not be adequate to characterize real-world scenarios.
3) Most of these algorithms require a large number of iterations to reach satisfying performance. The complexity of IoT systems makes optimization infeasible in polynomial time.

B. DRL Fundamentals and MDP

Fortunately, the recent advances in DRL have shown great potentials to address the 4Cs problems. As a subgroup of artificial intelligence (AI), DRL can be considered as an integration of reinforcement learning (RL) and deep learning (DL). On the one hand, RL involves self-learning agents that focus on maximizing the long-term performance, which learns from interactions with environments and requires no awareness of system models. On the other hand, DL has achieved remarkable progress on the challenges of high-dimensional sensory representation. Inspired by the biological neural networks, DL is constructed on top of the artificial neural network that typically has multiple layers. Theoretically, a multi-layer neural network with one or more hidden layers has the possibility of being a universal approximator [11]. Therefore, by leveraging DL, DRL is able to efficiently extract features and avoid the curse of dimensionality.

Essentially, DRL is applied for sequential decision-making, which can be mathematically formulated as an MDP [12], as shown in Fig. 2. An MDP is a discrete-time stochastic control process, where both the future state and reward of the environment only depend on the current state and the action taken. In general, an MDP can be defined by a tuple $(\mathbb{S}, \mathbb{A}, P, R, \mathbb{T})$:

- A finite set $\mathbb{S}$ of all possible states;
- A finite set $\mathbb{A}$ of all available actions;
- A transition probability function $P : \mathbb{S} \times \mathbb{A} \rightarrow \Delta(\mathbb{S})$ that maps from any state-action pair to the probability distributions of the next states;
- An immediate reward function $R : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$, where $\mathbb{R}$ denotes a finite set of all possible rewards;
- A sequence of time $\mathbb{T}$.

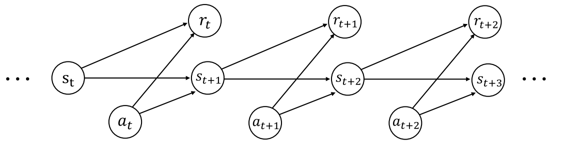


Fig. 2. An illustration of MDP model, where the subscript is the time step and an arrow is drawn from the dependent class to the corresponding dependency.

At decision epoch *t*, the agent observes the current environment state s_t and uses its policy to select an action a_t. A policy can be considered as a mapping from any state $s(s \in \mathbb{S})$ to an action $a(a \in \mathbb{A})$ (or action selection probability). After performing the action a_t, the environment moves to the next state s_{t+1} following the transition probability $P(s_{t+1}|s_t, a_t)$. In addition, a corresponding reward $r_t = R(s_t, a_t)$ is obtained via the immediate reward function, which is the evaluative feedback of the action taken. In an MDP, (s_t, a_t, r_t, s_{t+1}) is called experience. Given a stationary and Markovian policy π, the next state of the environment s_{t+1} is completely determined by the current state s_t. In this context, the current policy together with the transition probability function determines the long-term cumulative reward. Assuming $\tau = (s_t, a_t, s_{t+1}, a_{t+1}, \ldots, s_T, a_T)$ is a trajectory from an MDP, the long-term cumulative reward can be defined as:

$$G(\tau) = \sum_{i=0}^{T-t} \gamma^i R(s_{t+i}, a_{t+i}), \tag{1}$$

where $\gamma \in (0, 1]$ is the discount factor that measures the importance of future reward and *T* is the length of an episode. For continuous MDP, we have $T \to \infty$. In an MDP, the essential problem is to find the optimal policy that maximizes the long-term cumulative reward.

C. DRL Classifications

To meet the diverse needs of different IoT applications, varieties of DRL algorithms are emerging over the years. Depending on whether the algorithm emphasizes reward or policy, DRL algorithms can be classified as value-based DRL and policy-based DRL.

1) Value-Based DRL: In this paradigm, instead of storing an explicit policy, DRL algorithms attempt to approximate the value function, which is the estimation of the expected long-term reward of a state (or a state-action pair). Accordingly, the value function can be defined as a mapping from each state to its corresponding long-term reward, namely $V^\pi : \mathbb{S} \to \mathbb{R}$, where π is a policy derived from the value function. Mathematically, for all state $s_t \in \mathbb{S}$, the value function can be expressed as:

$$V^\pi(s_t) = \mathbb{E}_\pi[G(\tau|s_t)] = \sum_\tau P(\tau|\pi, s_t)G(\tau), \tag{2}$$

where $P(\tau|\pi, s_t)$ is the probability of trajectories that start with s_t and follow policy π, and $G(\tau)$ is the long-term reward defined in Equation (1). Formally, $V^\pi(s_t)$ is called the state-value function in RL. Among all state-value functions, there is an optimal state-value function that has the highest value for all states, i.e., $V^*(s_t) = \max_\pi V^\pi(s_t), \forall s_t \in \mathbb{S}$.

In addition, derived from $V^\pi(s_t)$, the action-value function (or *Q*-function) is introduced, which maps from any state-action pairs to the long-term rewards, namely, $Q : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$. The action-value function is particularly useful for focusing on a particular action at a particular state. The difference between the state-value function and action-value function is that the initial action is given in the action-value function. Mathematically, for all state $s_t \in \mathbb{S}$ and for all action $a_t \in \mathbb{A}$, the action-value function $Q^\pi(s_t, a_t)$ can be expressed as:

$$\begin{aligned} Q^\pi(s_t, a_t) &= \mathbb{E}_\pi[G(\tau|s_t, a_t)] \\ &= \sum_\tau P(\tau|\pi, s_t, a_t)G(\tau), \end{aligned} \tag{3}$$

where $P(\tau|\pi, s_t, a_t)$ is the probability of trajectories that start with (s_t, a_t) and follow policy π. And similarly to $V^*(s_t)$, the optimal action-value function is defined as $Q^*(s_t, a_t) = \max_\pi Q^\pi(s_t, a_t), \forall s_t \in \mathbb{S}, \forall a_t \in \mathbb{A}$.

In value-based DRL algorithms, the self-learning agents aim to obtain the optimal control policy π^* that excels all other policies in terms of the cumulative discounted rewards. Given the optimal value function $V^*(s_t)$, we can easily extracted the corresponding optimal policy as:

$$\pi^* = \arg\max_\pi V^\pi(s_t), \ \forall s_t \in \mathbb{S}. \tag{4}$$

Note that Equation (4) requires the use of environment model to obtain the optimal action of a state. To avoid this, many value-based DRL algorithms prefer approximating the action-value function $Q^\pi(s_t, a_t)$. Similarly, given the optimal action-value function $Q^*(s_t, a_t)$, the corresponding optimal policy can be easily extracted by:

$$\pi^*(s_t) = \arg\max_{a \in \mathbb{A}} Q^*(s_t, a), \ \forall s_t \in \mathbb{S}. \tag{5}$$

Therefore, in value-based DRL, the problem of searching the optimal policy is transformed into the problem of approximating the optimal action-value function. There are two mainstream approaches to achieve the goal, that is, Monte Carlo (MC) estimation and Temporal Difference (TD) learning. On the one hand, MC estimation periodically updates the action-value function and improves the policy according to the principle of generalized policy iteration [13]. This iterations can be represented as:

$$\pi_0 \to Q^{\pi_0} \to \pi_1 \to Q^{\pi_1} \to \cdots \pi_k \to Q^{\pi_k} \cdots \pi^* \to Q^*, \tag{6}$$

where π_0 denotes the initial policy. In particular, the action-value function is updated based on the averaging long-term rewards of historical experience. Given a full episode of experiences τ and the current policy π_k, for each state-action pair (s_t, a_t) appearing in the episode, MC method first calculates the long-term cumulative reward $G(\tau|s_t, a_t)$ with Equation (1), and then updates the current estimation $Q^{\pi_k}(s_t, a_t)$ with:

$$Q^{\pi_k}(s_t, a_t) \leftarrow Q^{\pi_k}(s_t, a_t) + \alpha(G(\tau|s_t, a_t) - Q^{\pi_k}(s_t, a_t)), \tag{7}$$

where α is the learning rate that controls the learning progress of models. With sufficient experiences, the action-value function will approach the actual action-value function with respect to the current policy asymptotically. Under this assumption, MC method improves the policy by constructing a new policy that is greedy with respect to the current action-value function. Specifically, for all $s_t \in \mathbb{S}$, the new greedy policy deterministically takes the action with the maximal long-term reward, which can be expressed as:

$$\pi_{k+1}(s_t) \leftarrow \arg\max_{a \in \mathbb{A}} Q^{\pi_k}(s_t, a). \tag{8}$$

According to the policy improvement theorem [13], the new greedy policy π_{k+1} is uniformly better than the previous policy π_k, unless π_k is the optimal. This ensures that the MC method will eventually converge to the optimal policy π^* and optimal action-value function Q^*.

On the other hand, TD learning follows a similar iterative learning process as in Equation (6), except that the update rule of action-value function is different. It uses the idea of Dynamic Programming and learns from incomplete episodes by means of bootstrapping. In particular, the action-value function is updated with the immediate reward and the estimated value of the next state. Mathematically, the update rule can be expressed in the following recursive form:

$$\begin{aligned} Q^{\pi_k}(s_t, a_t) &\leftarrow Q^{\pi_k}(s_t, a_t) \\ &+ \alpha\left(r_{t+1} + \gamma \max_{a \in \mathbb{A}} Q^{\pi_k}(s_{t+1}, a) - Q^{\pi_k}(s_t, a_t)\right), \end{aligned} \tag{9}$$

where α is the learning rate and γ is the discount factor. Compared with Equation (7), G_t is replaced with $r_{t+1} + \gamma \max_{a \in \mathbb{A}} Q^{\pi_k}(s_{t+1}, a)$, which is called TD target. Note that TD target exploits the Markov property, that is, the next state of the environment s_{t+1} is completely determined by the current state s_t. In this way, TD learning can learn online after every decision epoch, rather than waiting until the end of episodes. Typical value-based DRL algorithms include (deep) *Q*-learning [14], [15], SARSA [16], double deep *Q*-learning [17], etc.

Comparison between MC estimation and TD learning: MC estimation directly learns from the returns of complete episodes, thereby contributing to low bias in estimation. However, this also makes MC estimation require massive experience for convergence, suffer from high variance, and only be applicable for episodic MDP. TD learning, on the other hand, learns more efficiently, has low variance, and can be used to continuous MDP. However, TD learning is updated based on the prediction of next state, thereby having high bias and being more sensitive to initial action-value function. In general, the current DRL algorithms prefer to TD learning because it is more flexible and has stronger learning ability.

2) Policy-Based DRL: Next, we introduce another DRL paradigm, policy-based DRL. As shown in Equation (8), a policy in value-based DRL is inferred from the value function. In contrast, policy-based DRL algorithms target at modeling the policy and explicitly use parameterized approximators (such as DNN) to store policies. Let π_θ denote a policy with respect to parameters θ. In general, there are two kinds of policies: 1) deterministic policy that maps from each state to an action, i.e., $a = \pi_\theta(s)$; 2) stochastic policy that maps from each state to the probability distribution over the action space, i.e., $P(a|s) = \pi_\theta(a|s)$. Note that a deterministic policy can be interpreted as a stochastic policy by making the probability of the target action as 1 (and 0 for the remaining actions). Therefore, we focus on the stochastic case in the following.

In value-based DRL, symbols "V" and "Q" are typically used to denote the state-value and action-value. Similarly, policy-based DRL typically uses symbol "J" to denote the value/performance of a policy π_θ. Recall that the objective of DRL is to optimize the expected long-term reward, the objective function $J(\theta)$ can be mathematically defined as:

$$J(\theta) = \mathbb{E}_{\pi_\theta}[G(\tau)] = \sum_{\tau} P(\tau|\pi_\theta) G(\tau), \tag{10}$$

where $P(\tau|\pi_\theta)$ is the probability of trajectories following policy π_θ. Therefore, the essential problem of policy-based DRL is finding the optimal θ that maximizes $J(\theta)$, i.e., $\theta^* = \arg\max_\theta J(\theta)$.

For any differentiable parametrized policy π_θ, we can move θ towards maximizing the long-term reward by taking the derivative of $J(\theta)$ over θ. Based on the policy gradient theorem [18], we have:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_{\tau} P(\tau|\pi_\theta) G(\tau) \\ &= \sum_{\tau} P(\tau|\pi_\theta) G(\tau) \nabla_\theta \ln P(\tau|\theta) \\ &= \mathbb{E}_{\pi_\theta}[\nabla_\theta \ln \pi_\theta(a|s) G(\tau|s, a)], \end{aligned} \tag{11}$$

where $\mathbb{E}_{\pi_\theta}$ refers to the distribution of trajectories that follow the policy π_θ and $G(\tau|s, a)$ denotes the long-term reward of the trajectory τ that starts with (*s*,*a*). Equation (11) presents a general form of policy gradient.

However, calculating the expectation for all possible trajectories is typically intractable. With some simplifications, a famous policy-based DRL algorithm called REINFORCE [18] uses MC sampling to approximate the solution. Specifically, assuming that $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$ is a real sample trajectory following policy π_θ, for each state-action pair (s_t, a_t) in τ, MC sampling first calculates the long-term reward $G(\tau|s_t, a_t)$ with Equation (1), and then updates the policy parameters as follows:

$$\theta \leftarrow \theta + \alpha\gamma^t G(\tau|s_t, a_t) \nabla_\theta \ln \pi_\theta(a_t|s_t), \tag{12}$$

where α is the learning rate and γ is the discount factor. In this way, policy-based DRL moves parameters θ in the direction that yields higher reward.

Comparison between value-based DRL and policy-based DRL: Policy-based DRL uses parametrized approximators π_θ to model the policies, thereby providing better convergence properties than greedy policies and can learn stochastic policies. In addition, as shown in Equation (5), value-based DRL needs to enumerate all actions when making decisions, which is intractable for MDPs with high-dimensional/continuous action spaces. In contrast, policy-based DRL can avoid this by directly outputting the optimal action. However, policy-based

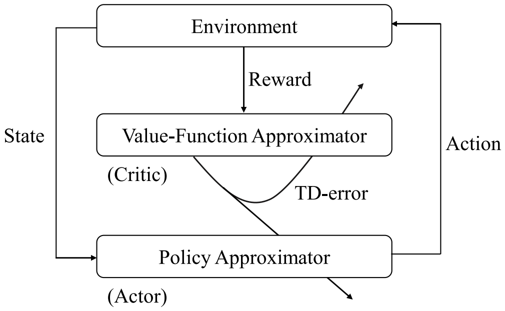


Fig. 3. The core elements of actor-critic architecture.

DRL tends to fall in local optima and is sensitive to parameter initialization. Moreover, policy-based DRL measures the performance of a policy through MC sampling, which results in high variability and noisy gradients.

D. Advanced Techniques in DRL

1) Actor-Critic Architecture: From the above analysis, we can learn that both the value-based DRL and policy-based DRL have their drawbacks. To address these challenges, the actor-critic architecture [19] was proposed by combing the strengths of value-function approximators and parameterized policies, which is regarded as a "hybrid" method. Fig. 3 shows its core elements, that is, an actor and a critic. The actor refers to the parameterized policy that takes a state as input and returns the optimal action or the probability distribution of all possible actions, while the critic refers to the value-function approximator that takes a state-action pair as input and determines its performance. In many DRL algorithms, the actor and critic are typically implemented by DNNs.

The principle of the actor-critic architecture is that the value-function approximator (i.e., critic) and parameterized policy (i.e., actor) complement one another to a large extent. On the one hand, the critic updates its parameters following rules similar to those of the value-based DRL. Let Q_ω denote the parameterized critic network with parameters ω and π_θ denote the parameterized actor network with parameters θ. Given an experience (s_t, a_t, r_t, s_{t+1}), the critic network is updated as:

$$\omega \leftarrow \omega + \alpha_\omega \delta_t \nabla_\omega Q_\omega(s_t, a_t), \tag{13}$$

where α_ω is the learning rate of critic network. In addition, δ_t is the TD error to correct the critic network, which can be computed with:

$$\delta_t = r_t + \gamma Q_\omega(s_{t+1}, a_{t+1}) - Q_\omega(s_t, a_t), \tag{14}$$

where a_{t+1} is sampled from the actor network, i.e., $a_{t+1} \sim \pi_\theta(a_{t+1}|s_{t+1})$. On the other hand, the actor network updates its parameters based on the weighted log-likelihood gradient estimations [18] suggested by the critic network. As mentioned earlier, the actor outputs the probability distribution of all possible actions. Therefore, the updating rule can be expressed as:

$$\theta \leftarrow \theta + \alpha_\theta Q_\omega(s_t, a_t) \nabla_\theta \ln \pi_\theta(a_t|s_t), \tag{15}$$

where α_θ is the learning rate of the actor. Note that there are many variations of actor-critic framework. Equation (15) is just one of the most commonly used updating rules. In short, the actor and critic complement one another and are together adjusted to maximize long-term rewards.

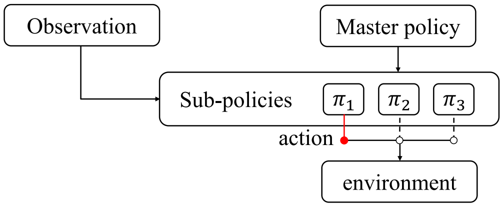


Fig. 4. An illustration of Hierarchical RL, where sub-policy π_1 is selected to output the environment action.

Advantages and drawbacks of the actor-critic architecture: By comparing with the value-based DRL, it is natural that the actor-critic architecture is more suitable for MDP problems with large/continuous action space. In addition, it can reduce variance and is more sample-efficient via the critic network. In fact, the actor-critic architecture has been widely adopted as the basic framework of later algorithms, such as Actor-critic [19], A2C, A3C [20], DPG [21], DDPG [22]. Nevertheless, the use of policy gradient makes it easy to fall into local optima.

2) Hierarchical RL: For complex IoT systems with sparse feedback or large action/state space, directly applying the above DRL algorithms may suffer from severe scaling issue. This is because DRL heavily relies on the reward signal to update the policy. To alleviate the training complexity, it requires DRL agents to identify the spatio-temporal abstractions and represent knowledge at a high level. Consequently, the hierarchical RL (HRL) framework was proposed by learning sub-policies with temporal or behavioral abstraction [23]. The core idea of HRL is to extend the available actions to macro-actions, which represent high-level controls over abstract objectives or long time scales. The high-level policies/controls determine the set of actions available in the lower-level. In particular, the lowest-level policies/controls output the environment actions as in traditional DRL algorithms. As shown in Fig. 4, the master policy selects sub-policy π_1 to output the environment action.

By dividing the original policy to multiple sub-policies, HRL brings several benefits. Firstly, decision-making problems are inherently combinable and hierarchical; this provides a potential approach for policy division [24]. Secondly, the structured exploration with different sub-policies contributes to a stronger generalization convergence ability, especially in environments with sparse rewards. This is because a lower-level policy can learn from the intrinsic rewards provided by the higher-level policy, rather than from sparse rewards provided by the environment. Thirdly, HRL makes it possible to apply the transfer learning between different sub-policies to achieve efficient learning effect [23].

3) Multi-Agent RL: Most of the above algorithms are built on MDP models, where the states of environments are fully available. However, in many real-world problems, some of the information required in the states may be private, or the collection of global information requires cooperation and communication among different agents, consequently resulting in

TABLE III
A LIST OF ADVANTAGES AND DISADVANTAGES OF DIFFERENT DRL ALGORITHMS

Methods	Advantages	Disadvantages	Examples
Value-based	- relative simple and has good performance - theoretical guarantees for globally optimal	- can not solve high-dimensional/continuous action spaces - can only learn deterministic policies	(deep) Q-learning [14], [15], SARSA [16], double deep Q-learning [17]
Policy-based	- better convergence properties - can learn stochastic policies - fit for high-dimensional/continuous action spaces	- tend to fall in local optima - sensitive to parameter initialization - high variance of gradient estimation	REINFORCE [18]
Actor-critic	- combine the strengths of value-based DRL and policy-based DRL	- tend to fall in local optima	Actor-critic [19], A2C, A3C [20], DPG [21], DDPG [22]

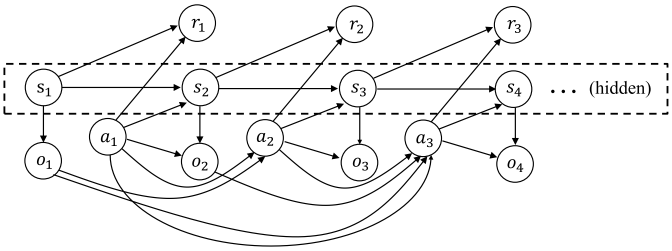


Fig. 5. An illustration of POMDP model. An arrow is drawn from the dependent class to the corresponding dependency and o_t denotes the observable information of the environment, where the subscript t is the time step.

high latency. In this context, traditional centralized approaches are infeasible for these Partially Observable MDPs (POMDPs). As shown in Fig. 5, the environment states are hidden and DRL agents can only make decisions based on its observations. The consequence of this unobservability is that POMDP is notoriously challenging to solve.

Multi-agent DRL (MARL) is an important branch of RL, which integrates game theory with RL and focuses on the long-term performance optimization in POMDPs [25]. To overcome the scaling issue, MARL distributes centralized control into multiple local agents. Each local agent is only responsible for the behavior of its local environment. Existing studies on MARL mainly focus on achieving efficient coordination between different RL agents and convergence to equilibrium. In general, MARL is technically and theoretically challenging. In addition to the challenges such as learning inefficiency as discussed above, there are new issues in MARL. For instance, each local agent may potentially cooperate, conflict, and influence other agents, thereby resulting in a constantly reshaped environment and non-stationarity of action performance. A good policy at the moment may not achieve the expected performance in the future caused by the updates of other agents. In consequence, it is challenging for MARL to achieve equilibrium between multiple agents. Nevertheless, MARL shows great potential in dealing with high-dimensional MDPs or POMDPs, especially in many IoT applications. Recently, some studies have tried to fully exploit the power of MARL to solve POMDP problems, such as mean field MARL [26] and MADDPG [27].

Summary: In this section, we have first described the 4Cs problems and the major research topics. Furthermore, to illustrate the advantages of introducing DRL into IoT domains, we have presented the core elements, fundamentals and classifications of DRL. Table III summarizes the advantages and disadvantages of different DRL algorithms. Different DRL algorithms can be used to address different problems in different IoT scenarios. In the following sections, we review the DRL applications on various problems in IoT scenarios.

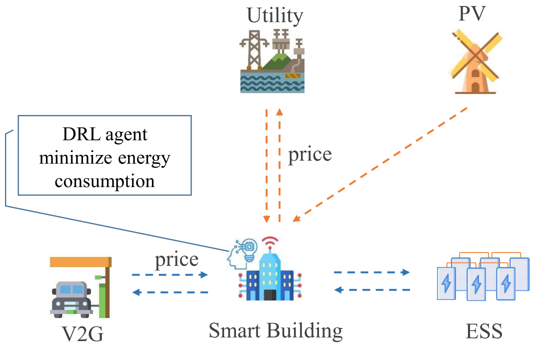


Fig. 6. Smart grid empowered smart building for minimizing energy consumption, where IoT devices provide real-time monitoring.

III. DRL APPLICATIONS ON IOT-ENABLED SMART GRID

The IoT-enabled smart grid that has been considered as the development trend of the electric system aims to construct a secure, sustainable, and energy-efficient electric system. The emerging DRL provides some plausible techniques for its future growth. In this section, we review the DRL applications on IoT-enabled smart grid and focus on the following issues: (i) energy management, (ii) demand response, (iii) wireless sensor network, and (iv) security and stability.

A. Energy Management

In smart grids, energy management enables precise strategy to improve energy efficiency, thereby filling the gap that conventional power grids lack of. Additionally, it uses advanced technologies to maintain load balancing and paves the way towards a sustainable, energy-efficient, and reliable power grid. To meet the above needs, the data-driven DRL offers new possibilities for developing intelligent energy management strategies, which are reviewed as follows.

1) Minimizing Energy Consumption: The recent advances in smart grids have witnessed new electricity generations and distributed energy resources, contributing to the shift from passive customers to active customers. During this procedure, DRL plays an important role in minimizing energy consumption. The authors in [28] proposed a smart building energy-saving algorithm based on deep *Q*-learning. As shown in Fig. 6, the system model consists of a smart building, a power grid that provides real-time power trading and supply, an energy storage system that provides storing and releasing ability, distributed renewable energy sources that help to reduce costs, and an vehicle-to-grid station (V2G) that manages the charging/discharging of electric vehicle (EV). The system is modeled as an MDP, where the model state s_t includes the current demand of the building and EVs, renewable energy generation, utility price in the power grid, and retained energy in the storage system. During each decision epoch, a deep Q-learning-based agent makes decisions a_t on whether to purchase or sell energy to the power grid, and whether to store or release energy from the storage system. In addition, the immediate reward r_t is the negative value of the cost paid for the energy. Given (s_t, a_t, r_t, s_{t+1}), deep Q-learning-based agent updates its action-value function following the recursive rule in Equation (9). The action-value function approximator $Q_\omega(s_t, a_t)$ of deep Q-learning is implemented as DNN, where ω denotes the parameters of DNN. Hence, $Q_\omega(s_t, a_t)$ is known as deep Q-network (DQN). By iterations, DQN $Q_\omega(s_t, a_t)$ is guaranteed to converge to the optimal value function. Simulation results based on real-life datasets show that the proposed algorithm achieves the lowest energy cost compared with greedy and random policies. Besides smart buildings, IoT-enabled smart grid also contributes to the development of energy harvesting devices [29].

Note that deep Q-learning-based algorithms [28], [29] can only deal with discrete action spaces. To quantify power distribution and energy transfer, an actor-critic-based algorithm called DDPG, was introduced in [30] for energy management. DDPG consists of two component: the critic network Q_ω as action-value function approximator and the deterministic actor network π_θ that directly outputs the target action, i.e., $a = \pi_\theta(s)$. Given an experience (s_t, a_t, r_t, s_{t+1}), the actor network is updated based on the deterministic policy gradient theorem [22], that is,

$$\theta \leftarrow \theta + \alpha_\theta \nabla_a Q_\omega(s, a)|_{s=s_t, a=\pi_\theta(s_t)} \nabla_\theta \pi_\theta(s)|_{s=s_t}, \quad (16)$$

where α_θ is the learning rate of actor network. Considering that the parameter updating of DNN may cause oscillation [15], DDPG implements additional DNNs for both the actor network and critic network, which refer to target actor network $\pi_{\theta'}$ and target critic network $Q_{\omega'}$. With a slight modification of Equation (14), the new TD error used to correct the critic network Q_ω is:

$$\delta_t = r_t + \gamma Q_{\omega'}(s_{t+1}, \pi_{\theta'}(s_{t+1})) - Q_\omega(s_t, a_t). \quad (17)$$

In addition, DDPG updates the target networks by slowly tracking the original network, that is,

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta', \tau \ll 1, \quad (18)$$

$$\omega' \leftarrow \tau\omega + (1 - \tau)\omega', \tau \ll 1. \quad (19)$$

It was shown that the introduction of DDPG achieves a significant improvement in on-line building energy optimization compared to deep Q-learning.

2) Economic Dispatch: The objective of economic dispatch is to determine the output of electricity generation and satisfy the customers demand at the lowest possible cost. Traditional approaches such as lambda iteration and interior point method were mostly built on power system models [31]. However, it is challenging to design accurate models for practical environments. Therefore, the authors in [32]–[34] designed economic dispatch algorithms based on deep Q-learning, which receive the power demand as system states and determine the allocated electrical power. The deep Q-learning-based algorithms interacts with the environment in an online manner, where the state transition probabilities and immediate reward function are unclear. In this case, the deep Q-learning-based agent learns the optimal policy through ϵ-greedy, that is, choosing action $\pi(s)$ with probability $1-\epsilon$ and choosing a random action with probability ϵ. In addition, the authors developed a distributed version of the algorithm, which relaxes the time constraints for collecting global information. Simulation results show that the algorithm can greatly improve the robustness and utility of the smart grid.

3) Balancing Energy Supply and Demand: Considering the time-varying customer demands and renewable energy generation of smart grids, the paradigm of the multi-agent system (MAS) provides an appealing approach to balance energy supply and demand. The authors in [35] proposed a multi-agent deep Q-learning model to investigate the distributed energy management problem in smart grids. Specifically, the system consists of multiple independent energy suppliers and consumers, which are modeled as interacting autonomous agents. Each agent is responsible for making local decisions based on its observation. Specifically, customer agents adjust or control the electrical load to minimize energy costs, and the supply agents allocate the electrical power to maximize their profits. The MAS converges to a Nash equilibrium when no agent can unilaterally change the policy to obtain higher profits, consequently leading to a stable system. Case studies in [35] show that compared with conventional approaches, the proposed algorithm can bring higher benefits to all agents when it converges to a Nash equilibrium.

Besides, MARL also contributes to the cooperation of large-scale integrated energy systems. The authors in [36] proposed a multi-agent deep Q-learning-based algorithm to maximize the utility of distributed generation and minimize grid operation costs. In view of multi-area smart generation problem, a MARL-based algorithm called Equilibrium $Q(\lambda)$ Learning was proposed in [37], which has superior performance and higher flexibility compared with single-agent algorithms. To effectively manage both renewable and traditional resources, a multi-agent deep Q-learning-based algorithm was proposed in [38], which coordinates power generation and storage to meet customer demands. However, partial observability, noisy observations, and indeterministic reward feedback increase the difficulty of MARL convergence, thereby limiting its application in larger-scale scenarios. This motivates us to design DRL agents that can adapt to such non-stationary.

B. Demand Response

Different from adjusting supply to meet energy requests, demand response (DR) shapes the demand profile of customers with approaches such as price-based DR to better match the power supply. For example, energy suppliers charge higher

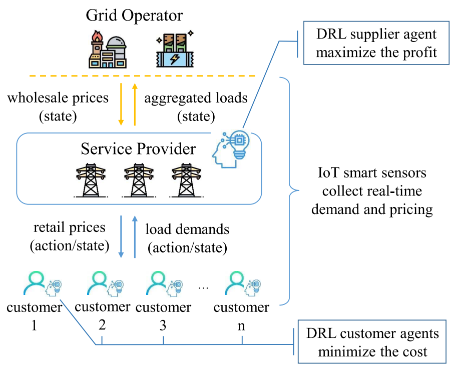


Fig. 7. A hierarchical electricity market model.

prices to reduce peak demand. By adjusting the flexible demands of customers, DR is capable of quickly taking measures to cope with supply-demand mismatches. In particular, the integration of DRL and DR revolutionizes conventional solutions and considerably stabilizes the smart grid by learning the optimal control policy.

1) Centralized Schemes: For centralized DR schemes, a central controller holds the global states and jointly considers the interests of both energy suppliers and customer [39]. Lu *et al.* [40] proposed a DR algorithm based on deep Q-learning to improve stability and reduce operation costs. As shown in Fig. 7, the power system is modeled as a hierarchical electricity market, which consists of a grid operator (GO), a service provider (SP), and several customers. IoT sensors are deployed to collect the real-time demand and pricing data in the electricity market. The central scheduler implemented in the SP makes a dynamic pricing strategy based on the aggregated load demands of customers and the wholesale electricity prices of GO to reduce energy consumption without jeopardizing smart grid stability. The global objective function of the central agent is the weighted sum of SP profits and customers costs, through which the proposed DR algorithm attempts to implement a win-win strategy for both sides. Similarly, Lu and Hong [41] proposed a real-time incentive-based DR framework to stabilize the IoT-enabled smart grid by purchasing energy from customers during peak times.

Different from using weighted reward functions, some studies seek to learn intelligent pricing strategies upon some specific customers response models. For instance, the authors in [42] proposed a deep Q-learning-based algorithm, which has enjoyed great performance improvements under various customers response patterns, including linear, logarithmic, and exponential models. Kim *et al.* [43] adopted an accumulated customer load demand model to investigate energy management and dynamic pricing in smart grids. However, the centralized scheme has the following disadvantages. Firstly, it is time-consuming to collect all the required information. Secondly, customers may try to hide the real responses with additional noise because of security and privacy concerns.

2) Distributed Schemes: Different from centralized schemes, distributed DR algorithms attempt to match energy demand and response by implementing distributed control agents. Wang *et al.* [44] developed a dynamic pricing strategy based on multi-agent Q-learning for the electricity trading market under incomplete information. In the model, prior to each trading, a market operator collects reservation bids and electricity demands from customer agents, as well as reservation prices and instant supplies from supplier agents. Next, the market operator adopts a double auction model proposed in [45] to simulate energy trading, which is extensively used to protect customer privacy. Among them, the double auction model is used to calculate the trading amount and price of customers and suppliers. After completing the trading, each agent receives feedback from the market operator and calculates its outcome with its own utility function. The authors in [44] have proved the existence and uniqueness of the mixed-strategy Nash equilibrium in their proposed repeated non-cooperative trading game. Experimental results show that the utilities of both customers and suppliers all converge to a desirable outcome with the proposed DRL-based DR scheme. Inspired by the previous work, Hurtado *et al.* [46] proposed to use distributed DQNs to enhance the stability of the smart grid while satisfying customer demands.

In addition to the tradeoff between profit and grid stability, each supplier in the smart grid also faces competitions from other coexisting suppliers. In response to this challenge, Wang *et al.* [47] proposed a multi-agent SARSA-based algorithms to manage the competitions among suppliers. SARSA is the on-policy version of Q-learning, and it selects the maximum achievable reward based on the policy that derives the previous action. Compared to Equation (9), the update rule of SARSA can be expressed as:

$$\begin{aligned} Q^{\pi_k}(s_t, a_t) &\leftarrow Q^{\pi_k}(s_t, a_t) \\ &\quad + \alpha(r_{t+1} + \gamma Q^{\pi_k}(s_{t+1}, a_{t+1}) - Q^{\pi_k}(s_t, a_t)), \end{aligned} \tag{20}$$

where is the selected action for the next state a_{t+1} and is performed at the next decision epoch. Simulation results indicate that the proposed algorithm can adapt to the dynamic energy trading market. The authors in [48] extended the on-policy DRL algorithm to an off-policy version, and developed a multi-agent deep Q-learning-based algorithm. In contrast, an off-policy DRL algorithm contributes to better exploitation and performance. However, the above studies fail to consider potential collusion and malicious attacks in the energy trading market. For instance, some distributed agents may collude and share their information to gain higher profits, consequently leading to a more complex MDP. Therefore, potential collusion and other malicious attacks shall be considered in future studies.

C. Wireless Sensor Network for Grid Monitoring

The major advantage of IoT-enabled smart grid is the ability to make effective decisions based on dynamically changing environments. To achieve this, a wide variety of wireless sensors have been widely deployed to realize real-time monitoring and evaluation of the grid status. However, the highly complex smart grid environments and resource-constrained wireless sensors pose great challenges in maintaining efficient operation of the system.

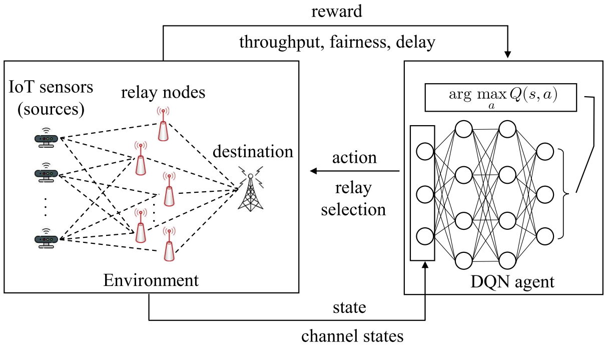


Fig. 8. The relay selection model in wireless sensor network.

1) QoS Communication Network: To provide QoS communication network, a relay node cooperation algorithm based on deep *Q*-learning was proposed in [49]. As shown in Fig. 8, the source node (i.e., IoT sensors) collects the global channel states as input and uses deep *Q*-learning to make an optimal relay selection decision (as action). For all output action-values, the deep *Q*-learning agent selects the action with the highest action-value as the target action, i.e., $\arg\max_a Q(s, a)$. And the reward function is defined as the weighted sum of throughput, fairness, and delay. Simulation results show that the proposed algorithm has better performance, lower failure rate, and lower energy consumption. To support ultra-dense wireless communications, an energy-efficient MAC protocol derived from deep *Q*-learning was proposed in [50], which shows higher performance in terms of throughput, fairness, and delay compared with the traditional CSMA-CA MAC protocol.

In large-scale wireless sensor networks, it is important to consider the transmission interference among different nodes, which leads to MDPs with large or continuous action space. To address the issues, Chinchali *et al.* [51] developed a network scheduling algorithm based on DDPG, which used a parameterized policy $\pi_\theta(s)$ to directly output scheduling decisions. Furthermore, the authors in [52] solved the challenge in another way. Specifically, an algorithm derived from multi-agent deep *Q*-learning was designed for multi-channel access problem, which was formulated as a POMDP. Each deep *Q*-learning-based agent takes the combination of historical chosen channels as the state s_t. Simulation results show that the proposed algorithm can adapt to the time-varying channel states and considerably reduce packet loss. In future work, the applications of MARL on more realistic and complex systems can be further investigated, such as multi-hop communication networks, where there is a tradeoff between communication efficiency and agent performance.

2) Energy Harvesting Strategy: To resolve the resource constraints and maximize the lifetime of wireless sensor networks, charging strategies have been extensively studied in recent years. The authors in [53] leveraged *Q*-learning to the charging-path scheduling problem, consequently improves the energy-efficient and prolongs the network lifetime. He *et al.* [54] extended the work in [53] to multi-hop scenarios and developed a path scheduling algorithm based on deep *Q*-learning to handle larger state space. On the other hand, the authors in [55] managed the operation modes of wireless sensors with DRL to adapt the time-varying energy supply of the external environment. In particular, the operation modes include sleeping, data collecting, processing, transmitting, and charging. The DRL agent controls the sensors based on their remaining battery capacity and external energy supply to extend network life and improve energy efficiency. However, the above studies do not consider the heterogeneity of wireless sensor networks. For example, there may exist different communication and charging protocols, which shall be considered in future works.

3) Power Control: Transmission power control is essential to prevent unwanted interference and improve energy efficiency. The authors in [56] introduced a deep *Q*-learning agent to develop policies that can adaptively adjust power based on network status. Specifically, the state is the interference intensities between sensor nodes, and the action is the transmission power level. Simulation results reveal that the proposed algorithm can improve the reliability and energy efficiency of the network. In addition, for large-scale wireless sensor networks, the authors in [57] addressed the transmission interference problem with a multi-agent deep *Q*-learning-based power control algorithm. The introduction of distributed decision-making agents reduces response time and makes it possible to apply the DRL technique in large-scale networks. Through simulations, the proposed algorithm outperforms the state-of-the-art power control algorithms. However, because of the uncertainty of the network environment, the proposed distributed algorithms require a large amount time to converge to a Nash equilibrium. In the future, techniques to accelerate convergence can be considered, such as common knowledge reinforcement learning, which learns hierarchical policy trees to coordinate decentralized tasks among multiple agents.

D. Smart Grid Security

The distributed architecture of IoT-enabled smart grids raises many security issues and privacy concerns, primarily involving data communication and sensor data collection. Cyber-attacks, false data injection, electricity theft or remote shut-off are the main threats to the smart grid. In view of this, we examine recent studies on grid security as follows.

1) Vulnerability Analysis and Defense: Vulnerability analysis is the process of identifying vulnerabilities, evaluating risks, and performing threat assessments. It is an important components of the security assurance mechanism of IoT-enabled smart grid. Yan *et al.* [58] proposed a *Q*-learning-based identification algorithm for power transmission sequential topology attack. To improve learning efficiency, the authors enhanced *Q*-learning with Monte Carlo Tree Search (MCTS) [59], which searches for optimal decisions in a given search space and moves to the most promising directions. In the model, the attacker uses a *Q*-learning agent to recognize the most vulnerable part in sequential attacks, while the smart grid with defense system works as an independent environment and takes measures responding to the attacks. Note that an attacker launches attacks with incomplete topological

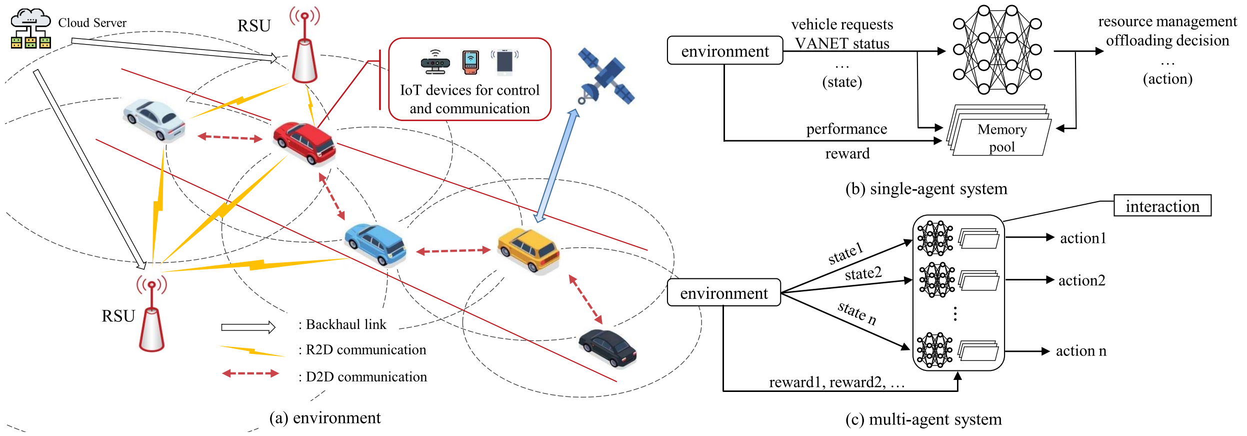


Fig. 9. (a) A typical IoT-enabled ITS environment that consists of vehicles, RSUs, and cloud servers. (b) A centralized DRL model in IoT-enabled ITS. (c) A typical distributed DRL model in IoT-enabled ITS.

information. Through trial-and-error learning, the Q-learning agent can trigger critical blackouts with the minimum number of attacks. In this way, we can identify grid vulnerabilities and prevent cascading failures caused by inter-connectivity.

In contrast to simulating attacks as in [58] and [60], some studies try to enhance grid reliability by implementing DRL control agents to obtain the optimal policies against various attacks. Duan *et al.* [61] mitigated the uncertainty of the grid with an autonomous voltage control algorithm based on deep Q-learning, which adaptively adjusts the generator setpoint based on the current power flows, voltage magnitudes and phase angles. Similarly, Wadhawan and Neuman [62] leveraged two value-based DRL algorithms, namely, deep Q-learning and SARSA, to devise the optimal resource allocation strategies for cyber attacks. Furthermore, the authors in [63] extended the model to MAS and modeled the interactions between attackers and defenders as a general sum stochastic game.

2) False Data Injection: Recent studies on smart grid security have shown that false data injection interferes grid status measurement and distorts reasonable operation, leading to blackouts or cascading failures [65]. From the perspective of attackers, Chen *et al.* [66] proposed to use deep Q-learning to implement automatic voltage control under false data injection. In this scheme, the attack-defence interactions are modeled as a POMDP, where a deep Q-learning attacker agent launches attacks based on limited information. Through the online trial-and-error process, the attacker agent can learn the optimal attack strategy, which allows us to implement corresponding countermeasures against such attacks. From the perspective of defenders, Kurt *et al.* [64] proposed using SARSA to detect potential attacks, which has a wider applicability compared with [66]. This is because no specific attacker model is required. In this context, it can detect unknown threats and disclose new types of attacks. Simulation results show that even a minor data deviation can be detected with the proposed algorithm. In the future, we can consider the false data injection problem from the perspective of both attackers and defenders, which involves MAS and game-theory. Because the environment is partially observable and the outcome of the policy depends both on its local environment and other agents, a coordination scheme that balances performance and communication overhead is required.

Summary: In this section, we have reviewed the DRL applications on IoT-enabled smart grid from different aspects, which are summarized in Table IV. We observe that owing to its simplicity, deep Q-learning method receives more attentions than other DRL methods in smart grid. Nonetheless, the applications of DRL on IoT-enabled smart grid are relatively theoretical. Most of the existing work is limited to the theoretical simulation. Therefore, it is necessary to study the utility of DRL algorithms under real-world settings in the future.

IV. DRL Applications on IoT-Enabled Intelligent Transportation Systems

Integrating advanced communication technologies, wireless IoT sensors and adaptive control strategies, IoT-enabled ITS makes it possible for a safer, more coordinated and efficient transportation network. In this section, we present some typical DRL applications on IoT-enabled ITS and discuss the existing challenges and future directions.

A. QoS in Vehicular Communication Networks

The need for QoS wireless communication in IoT-enabled ITS stems from the time-varying dynamics, high density, and security requirements of vehicular networks. Fortunately, recent advances in communication technology greatly contribute to the success of many ITS applications in Vehicular Ad-hoc Networks (VANETs). Fig. 9(a) shows a typical VANET model, which consists of: 1) vehicles equipped with IoT devices that provide intelligent control and communication capabilities; 2) Road Side Units (RSUs) that provide additional computational and communication resources; 3) cloud servers that connected to RSUs and provide computational supports. In the following, we present some recent advances in utilizing DRL for QoS wireless communication.

1) Network Resource Optimization: To address the high mobility of VANETs, Roadside Units (RSUs) with wireless connectivity supports are widely deployed in VANETs. In

TABLE IV
DRL APPLICATIONS ON IOT-ENABLED SMART GRID

Algorithms	Applications	4Cs problems	MDP structure			Refs.
			State	Action	Reward	
Deep Q-learning	reducing energy consumption	- control	- demand, storage - unit price	- buy/sell - store/discharge	- energy cost	[28], [29]
	economic dispatch	- control	- energy demand - generation cost - capacity constraints	- unit commitment	- electricity purchased	[32] – [34]
	improving energy-efficiency	- control	- demand, storage - wholesale price	- trading amount - price	- net profit	[40] – [43]
	QoS control in network	- communication - caching - control	- network status - user requests, etc.	- relay node selection - channel selection, etc.	- energy consumption - throughput	[49] – [55]
	interference prevention	- communication - control	- user interference - network status, etc.	- transmission power	- energy efficiency - interference level	[56], [57]
	smart grid security	- communication - control	- grid status	- sequential attacks - voltage control	- vulnerability analysis - detection sensitivity	[58] – [62]
Multi-agent deep Q-learning	balancing energy supply and demand	- control	- energy demand - unit price	(customers) - submitted bits	- energy cost	[35] – [38]
			- demand, storage - generation cost	(suppliers) - energy sold	- net profit	
	demand response	- control	- energy demand - retail price	(customers) - energy purchased	- energy cost	[44] – [46]
			- demand, storage - wholesale price	(suppliers) - price, amount	- net profit - grid stability	
Multi-agent SARSA	demand response	- control	- energy demand - retail price	(customers) - energy purchased	- energy cost	[47], [48]
			- demand, storage - wholesale price	(suppliers) - price, amount	- net profit - grid stability	
	defence against false data injection	- communication - control	- grid status	- inject false data	- grid damage level	[62], [64]
				- detect false data	- detection sensitivity	
DPG/DDPG	reducing energy consumption	- control	- demand, storage - unit price	- energy purchased	- energy cost	[30]
	QoS control in network	- communication - control	- network status - user requests	- channel selection	- energy consumption - throughput	[51]

general, from the perspective of energy saving, the resource-constrained RSUs prefer providing services with vehicles that are relatively close. However, it can lead to incomplete services and poor user experience. Accordingly, Atallah *et al.* [67] presented an intelligent RSU scheduling algorithm based on *Q*-learning, being implemented in centralized manners, as shown in Fig. 9(b). The *Q*-learning agent receives vehicle requests and system states as input, then selects vehicles to serve as output. Additional penalties for incomplete services are added to the rewards for high user experience. To improve data efficiency, the agent stores the transition (s_t, a_t, r_t, s_{t+1}) in the replay memory after each interaction. Then, to update the parameters of DQN, the agent randomly selects a set of independent transitions as a minibatch. In this context, the loss function can be expressed as:

$$L(\omega) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1} \sim \rho(\cdot))}[\delta_t], \tag{21}$$

where $(s_t, a_t, r_t, s_{t+1} \sim \rho(\cdot))$ represents experience sampled from the replay memory and δ_t is the TD error defined in Equation (14). Simulation results show that the proposed algorithm can increase the number of vehicles being served and avoid incomplete service. Different from [67], Yan *et al.* [68] extended the above work to MAS, where multiple RSUs cooperate with others to satisfy user demands and maximize VANET throughput. Using MARL, RSUs can provide greater utility and cover a larger area.

Spectrum allocation is another major challenge for IoT-enabled ITS; it is mainly caused by the high mobility and high density of VANETs. Firstly, to solve the inherent high mobility, radio spectrum needs to be switched frequently, which results in the unpredictability of available spectrum. To address this issue, MARL is introduced, which distributes centralized control into multiple local agents. As shown in Fig. 9(c), each local agent is only responsible for the behaviour of its local environment and outputs action based on its local states. For instance, the authors in [69] proposed a multi-agent deep *Q*-learning-based algorithm to coordinate the joint spectrum of cascaded base stations. In this model, each base station agent takes actions based on the states of itself and its neighboring base stations. Compared with single base station spectrum management, multi-agent cooperation can detect spectrum occupancy and reduce switching frequency. Secondly, network function virtualization provides a flexible framework to allocate network resources. In [70] and [71], Fu *et al.* decomposed the complicated IoT network into multiple small virtual network function components. In addition, a deep *Q*-learning-based algorithm is developed, which can greatly improve the spectrum utility as well as user experience.

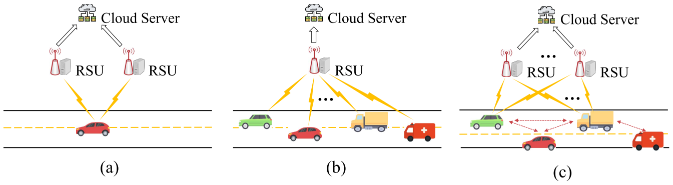


Fig. 10. Different computation offloading model in VANET: (a) one-to-many, (b) many-to-one, (c) many-to-many.

Regarding the path scheduling, a routing scheme called QGrid was proposed in [72] by combining multi-agent Q-learning with grid-based routing. Because the geographical area in the system can be split into many girds, multiple agents can be applied to speed up convergence, where each agent decides which grid to move forward data. Although the movement of vehicles is highly dynamic, it exhibits a certain pattern in the macroscopic [73]. Therefore, the objective of multi-agent Q-learning is to derive the patterns and determine the optimal path from source to destination. Simulation results based on real-life data validate that QGrid can improve network throughput. In addition, the authors in [74] integrated transfer learning with deep Q-learning, which greatly improves the learning efficiency of DRL agents.

Another key enabling technology for Vehicle-to-Vehicle (V2V) communication is interference management, which are described as dynamic games between distributed agents [75]. Challita *et al.* [76] proposed a decentralized Q-learning scheme with latency constraints to reduce interference without restricting the autonomy of vehicles. Simulation results show that the proposed algorithm can converge to a subgame perfect Nash equilibrium (SPNE). Extended from the previous work, Challita *et al.* [77] introduced an Echo State Network (ESN) architecture [78] to predict future environment conditions. Subsequently, DRL agents make decisions based on the predictions from ESN. Simulation results show that this improvement can greatly speed up the training process.

The growing number of IoT devices and the diversification of applications yield the heterogeneity, autonomy, and large scale in the vision of VANETs. Although DRL has enjoyed tremendous success in many fields, it may face some technical challenges in the future VANETs because of the time-consuming training process. Therefore, the need for an efficient and robust network resource allocation algorithm that can provide real-time decision-making and quickly adapt to the dynamic VANETs sheds light on future research.

2) Communication Security: For VANETs, security attacks such as eavesdropping, jamming, and spoofing can lead to insecure transmissions, packet loss, or even catastrophes. Considering the diversity of attack models and the autonomy of attackers, the authors in [79] focused on enhancing the transmission security by adaptive power control. The system is formulated as a noncooperative game between data transmitters and malicious attackers. A Q-learning agent is applied to allocate the transmission power over multiple channels without the knowledge of channel and attacker models. Simulation results show that the proposed algorithm can greatly restrict the advantages of attackers in the game, thereby suppressing the attack. Inspired by the aforementioned work, the authors in [80] developed a power allocation algorithm based on deep Q-learning. This scheme exploits beamforming against eavesdropping and receiver filters against spoofing.

Anti-jamming is another challenging problem in communication security, which involves radio noise that deliberately causes transmission failures. This is different from the interference where radio noise is unintentional. In this context, the anti-jamming problems are typically formulated as POMDPs. Xiao *et al.* [81], [82] applied multi-agent deep Q-learning to achieve anti-jamming power control. Through interacting with intelligent attackers, defender agents on the vehicles determine the transmit powers and relay nodes. The state consists of the link qualities, signal-to-interference-plus-noise-ratio (SINR), and bit-error-rate (BER). The proposed algorithm is attractive because it greatly improves network utilization and reduces the BER of messages.

B. Edge/Fog Computing for ITS Applications

In recent years, the conflict between the proliferation of highly demanding applications and the resource constraints of IoT devices has given rise to edge/fog computing, which provides a promising tool for moving computational resources to the edge of VANETs [83]. Fig. 10 shows three computation offloading models: (a) one vehicle selects the best RSU to offload tasks, (b) multiple vehicles complete for the resources in one RSU, (c) multiple vehicles complete for the resources in multiple RSUs. We review the DRL applications on IoT-enabled ITS under different offloading models as follows.

1) Integrated Networking, Caching, and Computing: In edge/fog computing, integrated networking, caching, and computing has been recognized as the key enabling technology. The authors in [84], [85] jointly considered the networking, caching, and computing in the computation offloading problem. The system model is shown in Fig. 10(c), where tasks are offloading through the device-to-device (D2D) communication network. Because of the large number of vehicles, the state dimension and the action dimension are extremely high. To address the issue, the authors developed a novel integrated framework based on dueling double Q-learning, which achieves an intelligent allocation strategy over networking, caching, and computing. Simulation results demonstrate the effectiveness of the proposed algorithm. Extended from the previous work, Yao and Ansari [86] developed a novel mobility-aware DRL algorithm, which reduces the variance of policy gradient with reward reshaping. Simulation results show that the proposed algorithm can quickly adapt to the changes of the system and obtain optimal policies in different environments. DRL applications for integrated networking, caching, and computing generally involve high-dimensional continuous state and action spaces, which is worth investigating in the future. Approximate nearest-neighbor method with candidate action set [87] or action elimination deep Q-network [88] provide promising prospects to address the challenges.

2) Load Balancing: The increasing offloading requests in IoT-enabled ITS drive the demand for a load balancing strategy in edge nodes. Driven by this challenge, Sen and Shen [89] proposed a Q-learning-based load balancing algorithm called RILTA, which enables high-efficient execution of offloading tasks. The system model is shown in Fig. 10(b). In the system, a RSU governs a set of edge nodes and collects vehicle requests as states. Compared with two state-of-the-art task assignment algorithms proposed in [90] and [91], RILTA can reduce energy consumption by 13 – 22% and task execution time by 1 – 10%. In addition, Wei *et al.* [92] proposed an actor-critic-based task scheduling algorithm for the joint optimization of communication, caching, and computing. These works provide enlightening insights into the promising yet underexplored domain of edge/fog computing. For future work, the time-sensitivity, security, and scalability of edge/fog computing need to be considered.

C. Real-Time Systems

One way to improve the user experience in IoT-enabled ITS is to deploy real-time systems, which can reduce the user-perceived latency and build a safer and more intelligent ITS. In addition, Markov-Chain has been widely used as an effective tool to measure the performance of real-time systems. In light of this issue, the following closely examines recent studies on DRL applications on real-time systems.

1) Traffic Signal Control: Spawned by the increasing volume of traffic, traffic congestion has become a major issue in ITS. Traditional rule-based traffic signal control algorithms fail to consider the dynamic changes of environments. In [93], a deep Q-learning-based framework was proposed for two tasks: traffic flow prediction and vehicle motion control. The framework performance is measured by the weighted sum of queue length, waiting time, light switches, and intersection throughput. A DRL-based partially detected intelligent transportation framework was further proposed in [94], where only vehicles equipped with IoT communication/sensing devices and within communication/sensing ranges are observable, as shown in Fig. 11. The proposed framework yields impressive performance results under extensive simulations of different traffic flows, network topologies, and detection ranges. However, when applied the above methods, huge communication overhead may occur because of the need for frequent data collection.

To overcome the scalability challenge of centralized approaches, Chu *et al.* [95] distributed the central agent to multiple independent local agents. By adaptively adjusting the reward signals received from other agents according to the distance, the proposed algorithm can considerably stabilize the convergence, thereby realizing faster coordination between distributed agents. A novel framework was further proposed in [96], where motorized traffic (such as vehicles) and non-motorized traffic (such as pedestrians) were both taken into account. In this case, the problem is quite complicated when considering the mobility of pedestrians. To overcome the high complexity of the problem, each local agent exchanges its environmental observations with its neighboring agents. The simulation is run under real-world maps and traffic data, the results of which show that the proposed algorithm can achieve higher performance in terms of queue length, waiting time, fuel consumption, and gas emission.

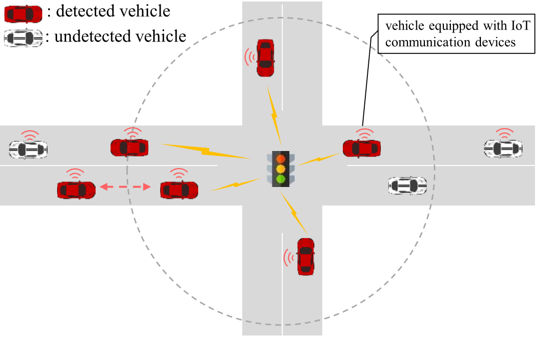


Fig. 11. Illustration of partially detected intelligent transportation framework.

2) Real-Time Data Acquisition: Because accurate traffic information is the foundation of many ITS applications, real-time data acquisition is another key enabling technology for future ITS. There have been several important attempts on Unmanned Aerial Vehicle (UAV) control for real-time data acquisition. Liu *et al.* [97] proposed a DRL-based UAV control algorithm called $\mathrm{DRL}-\mathrm{EC}^3$ to achieve optimal performance in terms of energy consumption, communication coverage, coverage fairness, and network connectivity. Toward this end, the UAV control problem is formulated as a joint optimization problem and the immediate reward function is a weighted sum of the above factors. In addition, Liu *et al.* [98] went one step further and implemented the previous $\mathrm{DRL}-\mathrm{EC}^3$ algorithm in a fully-distributed manner for a more realistic scenario. Instead of being control by a central agent, each UAV has an independent and autonomous decision-making agent. The benefits of a distributed scheme are twofold. First, it is preferable for large-scale systems because communication delays can be avoided. Second, it can be designed for parallelism, which allows it to train network models in parallel, thus accelerating convergence. Simulation results demonstrate the superiority of the distributed $\mathrm{DRL}-\mathrm{EC}^3$ algorithm compared with the centralized scheme and DDPG algorithm.

D. Autonomous Planning and Control

Autonomous planning and control contribute to the well-functioning of IoT-enabled ITS in many ways, such as congestion mitigation and collision avoidance. The mainstream practice is to decompose the entire autonomous control task into several high-level sub-tasks, such as speed limit control and vehicle lateral control. We next take a closer look at the DRL applications on autonomous planning and control as follows.

1) Speed Limit Control: For congestion mitigation at the freeway, Li *et al.* [99] proposed a Q-learning-based speed limit control algorithm. During the training process, the authors used a modified cell transmission model to simulate dynamic traffic. With the proposed algorithm, the freeway model can keep operating close to its maximum capacity. However, the above study fails to consider that the speed limit control system is a typical MAS. The development of VANETs

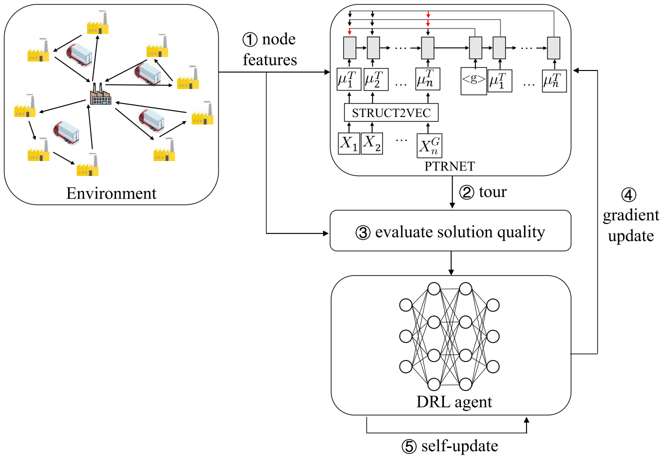


Fig. 12. An illustration of pointer network that is trained by DRL agent for online VRP.

makes it possible to share data between distributed agents. For instance, a speed limit control algorithm based on multi-agent deep *Q*-learning was proposed in [100], where multiple agents are distributed across different sections of the freeway. To reduce computational complexity, each agent assumes that the other agents take actions based on optimism, that is, taking the most beneficial actions. Meanwhile, the adjacent sections of the freeway can reduce their speed differences, leading to a lower collision risk rate. However, the above studies are built on the assumption that each vehicle is controlled by an intelligent speed limit control agent, which is impractical. Thus, the automated driving system still has a long way to go.

2) Vehicle Lateral Control: Another great challenge in the design of autonomous vehicles is the vehicle lateral control, including lane selection and lane-changing. You *et al.* [101] modeled the stochastic vehicle control problem in a multi-lane highway as an MDP, where available actions include accelerating, braking, maintaining, lane-keeping, left-turn, and right-turn. As a solution to this problem, a vehicle lateral control framework based on multi-agent deep *Q*-learning was proposed. Simulation results show that the proposed algorithm can realize cooperative lateral control and achieve maximum road throughput without collision after convergence. However, the proposed algorithm applied ϵ-greedy exploration strategy, which is based on the random mechanism and may output unsafe actions during the training process. Therefore, a deep *Q*-learning framework incorporating safety verification was proposed in [102], where each vehicle respects a safe distance from other vehicles. In this context, the DRL agent can guarantee collision-free control even at the beginning of the training process.

3) Online Vehicle Routing: Online vehicle routing is expected to be an integral component of modern transportation services, such as last-mile delivery. Although some recent studies have investigated the performance of many representative mathematical programming-based approaches in the vehicle routing problem (VRP), there remains a research gap in the online VRP. This is because the strict response time requirements of online VRP make it impractical to apply traditional mathematical programming-based approaches. Fortunately, deep learning can aid in solving the challenge by mapping directly from system states to routing solutions. As shown in Fig. 12, Yu *et al.* [103] designed a pointer network with LSTM cells and STRUCT2VEC embedding to develop vehicle routing solutions. In contrast to supervised learning, DRL does not rely on constructing a large data set explicitly with optimal solutions, which is generally extremely time-consuming. Instead, DRL uses a reward function that encourages the pointer network to output better solutions, which is the gradient update in Fig. 12. In addition, to reduce the computational complexity of network parameter optimization, the authors leveraged a representative DRL algorithm called A3C, which enables parallel training among multiple DRL agents with threads. Let $\{\pi_{\theta_1}, \pi_{\theta_2}, \ldots, \pi_{\theta_N}\}$ denote the thread-specific DRL agents and π_θ denote the global agent. A3C accumulates gradients over multiple agents to improve training efficiency, where the accumulation step is as follows:

$$d\theta \leftarrow d\theta + \nabla_{\theta_n} \ln \pi_{\theta_n}(a_t|s_t)(G(\tau|s_t) - V(s_t)), \quad (22)$$

where $G(\tau|s_t)$ is the long-term rewards defined in Equation (1), $V(s_t)$ is the approximate value, and θ_n is a policy sampled from $\{\pi_{\theta_1}, \pi_{\theta_2}, \ldots, \pi_{\theta_N}\}$. The accumulated gradient $d\theta$ is used to update the global agent asynchronously. Then, A3C synchronizes thread-specific agents with the global one as: $\theta_n \leftarrow \theta$. The performance of the proposed algorithm is verified by extensive case studies. However, this work is still elementary. Future work should further consider environments with more vehicles and investigate how to improve the solution quality.

Summary: In this section, we have reviewed the DRL applications on IoT-enabled ITS. Table V gives a summary of the related studies. We note that recent studies on the DRL applications on IoT-enabled ITS are mostly distributed and concerned with MARL, which is caused by scalability issues, communication delays and data synchronization. Nonetheless, many challenges remain in developing a MARL framework with fast convergence and stable performance. This is due to the lack of effective coordination mechanisms between distributed agents, which shall be further studied in the future.

V. DRL Applications on Industrial IoT

IIoT is envisioned to become the main driving force to reshape the future smart industry with provision of unprecedented data services. Industrial organizations and enterprises can benefit from the IIoT, which helps them save time, reduce costs, and make effective decisions. Driven by this, DRL is adopted to empower traditional industries from many respects, such as providing efficient communication and optimal resource allocation for IIoT systems.

A. Resource Optimization in Industrial Wireless Network

The realization of IIoT relies on seamless interaction between a large number of heterogeneous connected sensors, which calls for network resource optimization strategies in wireless industrial networks. Although industrial wireless networks share many common features with traditional

TABLE V
DRL APPLICATIONS ON IoT-ENABLED INTELLIGENT TRANSPORTATION SYSTEM

Algorithms	Applications	4Cs problems	MDP structure			Refs.
			State	Action	Reward	
(Deep) Q-learning	QoS control in VANET	- communication - control	- vehicle status - user requests, etc.	- vehicle movement - relay selection	- user experience - throughput, etc.	[67], [72] – [74]
	network security	- communication - control	- vehicle status - user requests, etc.	- power control - channel selection	- security analysis	[79], [80]
	integrated networking, caching, and computing	- communication - caching - computing	- computing/network resources - user requests, etc.	- networking, caching, and computing decision	- edge/fog computing performance	[84], [86]
	load balance	- computing - communication	- computing/network resources - user requests, etc.	- task assignment	- edge/fog computing performance	[89] – [91]
	traffic signal control	- control	- traffic status	- traffic signal	- traffic throughput	[93], [94]
	congestion mitigation	- control	- traffic status - vehicle status	- speed limit control	- roadway efficiency - driver experience	[99], [102]
Multi-agent deep Q-learning	QoS control in VANET	- communication - caching	- vehicle status - user requests	- relay selection - spectrum allocation	- throughput - spectrum utility	[75]
	interference avoidance	- communication	- vehicle status - user requests, etc.	- path scheduling	- VANET latency	[75] – [78]
	anti-jamming	- communication	- VANET status	- power control - channel selection	- jamming analysis	[81], [82]
				- jamming attacks		
	traffic signal control	- control	- observable traffic status	- traffic signal	- traffic throughput	[95]
	congestion mitigation	- control	- observable traffic and vehicle status	- speed limit control - lateral control	- roadway efficiency - driver experience	[100], [101]
Actor-critic	load balance	- communication - computing	- computing resources - user requests, etc.	- task assignment	- edge/fog computing performance	[92]
Multi-agent DDPG	energy-efficient data collection	- communication - caching	- data distribution - sensing area, etc.	- UAV movement - task assignment	- energy-efficiency, - fairness, etc.	[104]
A3C	routing strategy	- control	- node features	- routing	- routing cost	[103]

IoT network, they are more keen on the system reliability, performance and energy-efficiency [105], which will be highlighted in the following.

1) Energy-Efficient Resource Allocation: One of the major concerns of the industrial wireless network is energy-efficiency. This is due to the huge burden of frequent data generation, collection and exchange. Xu *et al.* [106] proposed a radio-resource allocation algorithm based deep *Q*-learning for the Cloud Radio Access Network (C-RAN), where C-RAN is a novel network architecture that enables massive data communication. The proposed algorithm enables flexible radio assignment in the C-RAN and greatly reduces energy consumption across the full operational period. Furthermore, Fan *et al.* [107] jointly considered the transmission power control and channel selection in large-scale wireless networks. In particular, the authors considered a time-varying wireless communication network and introduced the deep *Q*-learning agent to learn the optimal control strategy. Simulation results indicate a significant increase in energy efficiency with the proposed algorithm.

Note that the aforementioned studies only consider wireless networks from one communication mode. However, the device heterogeneity and resource diversity may lead to a more complex network where different communication modes coexist. Driven by this issue, the authors in [108] investigated the resource allocation problem in a dynamic, multi-resource, and multi-mode communication wireless network. They developed a model selection and resource management framework based on deep *Q*-learning to adapt to the dynamic network state. More explicitly, they depicted a system model that consists of a cloud center and multiple local devices equipped with D2D transmitters. In addition, they consider two communication modes: D2D mode and C-RAN mode. In C-RAN mode, real-time network virtualization are required and thus the energy consumption caused by the active processors in the cloud center is unavoidable. In contrast, D2D mode allows the cloud center to turn off some processors for energy saving. The deep *Q*-learning-based network controller quickly decides the communication mode of each local device and the on-off states of processors. Simulation results show that the proposed algorithm contributes to lower energy consumption without violating the transmission rate constraints.

2) Satisfying the Latency Constraints: Compared with traditional IoT systems, IIoT is more concerned with communication latency. This is because IIoT relies heavily on timely data acquisition to guarantee real-time control. In particular, inaccurate or incorrect decision-making caused by data collection latency can result in system failures and outages, which in turn can lead to life-threatening situations in real-world scenarios [109]. In a pioneering work [110], *Q*-learning was integrated into the channel selection strategy in the context of Cognitive Radio Networks (CRNs) and Spectrum Handoff (SH). SH is a dynamic spectrum access technique that can provide fair resource allocation, efficient channel utilization,

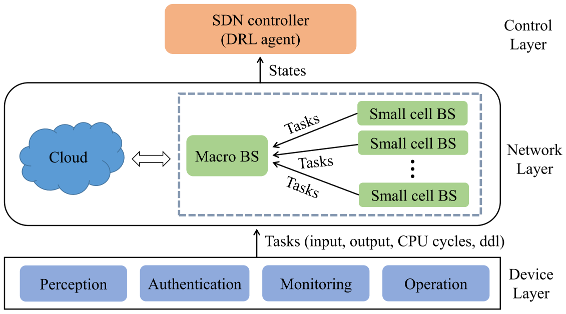


Fig. 13. A SDN-enabled multi-layer heterogeneous framework for IIoT systems.

and reliable real-time connectivity through cognitive radios. However, in the case of poor channel selection strategy, SH may instead consume extra network resources. Therefore, the authors in [110] proposed a Q-learning-based channel selection algorithm to minimize communication latency. In addition, a linearly-dependent masking scheme is introduced to speed up the convergence of Q-learning. Simulation results show that the proposed method significantly can reduce latency and improve throughput compared with traditional SH schemes.

Note that the aforementioned study fails to consider the conflict between the limited storage capacities of network nodes and the ongoing growth of data traffic. Packet loss may occur if the buffer for incoming packets is full. To address the challenges, Zhu *et al.* [111] discussed the transmission scheduling in a CRN-based IIoT. They designed a novel Q-learning-based scheduling mechanism that selects appropriate channels to transmit packets in different buffers. Then, they used a comprehensive action evaluation of Q-value and the index value for action selections, both of which ensure the balance between exploitation and exploration. Simulation results indicate a decrease in the packet loss rate of the proposed algorithm. However, the above studies leave many open questions for further study. For example, the exploration phase of DRL training is so long that online applications of DRL are almost infeasible. In addition, the resource allocation in IIoT systems typically involves multiple objectives, leading to a great challenge in designing reward functions.

B. Computation Offloading for Industrial Data Processing

As stated earlier, the full realization of intelligent manufacturing heavily relies on real-time control. However, the prosperity of IIoT applications has proposed higher requirements for both networks and devices [112]. Fortunately, computation offloading is emerging as an effective technique to provision scalable resources. Therefore, we next review DRL applications on computation offloading.

1) Optimizing Offloading Strategies: Designing an effective offloading strategy involves multiple domains, such as power control, workload distribution and resource allocation. For example, Wang *et al.* [113] proposed a deep Q-learning-based resource allocation scheme, which can adaptively allocate both computational and network resources in the edge servers to minimize the average service time. The proposed algorithm can still achieve satisfying performance under the pressure of burst requests. In addition, the authors in [114] considered a more practical scenario where the lengths of transmission queue and task queue are finite, and proposed a deep Q-learning-based algorithm for strategy optimization. In the model, the system state is defined as the information on requested devices, including the transmission queue size, buffer size, scheduling epochs, and edge computing priority. Meanwhile, the feasible action space that depends on the current state can be calculated from the cumulative Cartesian products of the action space of all tasks. Simulation results show that the algorithm improves the system performance in the aspects of task completion reward and execution time.

In contrary, some recent studies consider a novel network model where edge computing and cloud computing are mutually reinforcing. For example, Wang *et al.* [115] considered the computation offloading in an edge-cloud IIoT network and proposed a multi-layer heterogeneous computing framework as shown in Fig. 13. The framework combines different computational resources providers including a cloud center, a macro base station (MBS), edge servers and small cell base stations (SBSs). In addition, they defined the incoming task with four items, that is input size, output size, required CPU cycles, and maximum tolerable delay. In the model, a software-defined networking (SDN) controller is equipped to collect states and control task flow. More importantly, a deep Q-learning-based agent is deployed at the SDN controller to make intelligent decisions on task assignments such as staying in SBS, offloading to MBS and offloading to the cloud. The reward is defined to judge whether the action can lead to an increment in the task completion rate. Simulation results show that the proposed algorithm can achieve a close performance to the enumeration algorithm.

Furthermore, the authors in [116] put their emphasis on an edge-cloud IIoT environment with wireless charging devices. This scheme prolongs the battery lifetime and contributes to a lower operational cost. However, wireless charging may consume extra network resources. To address the issue, the authors proposed a deep Q-learning-based online offloading algorithm, which improves the long-term performance up to 56% compared with the greedy algorithm. However, the naive searching policy of deep Q-learning, namely ϵ-greedy algorithm, prevents its success in large-scale IIoT systems. Therefore, Chen *et al.* [117] improved the searching policy of tradition DRL algorithms with the MCTS algorithm, as illustrated in Fig. 14. A major contribution of introducing the MCTS algorithm is that it can generate training data from simulations and train the DRL agent via self-supervised learning. Let v_i denote a node in the Monte Carlo tree. MCTS expands the best child node of v_i with:

$$v_{i+1} \leftarrow \underset{v \in \text{child of } v_i}{\arg\max} \quad F(v), \tag{23}$$

where $F(\cdot)$ is a function measuring the performance of child nodes. Note that $F(\cdot)$ is determined by the environment. This work provides a promising approach to scale DRL agents to adapt to the ongoing growth of network scales. Future work

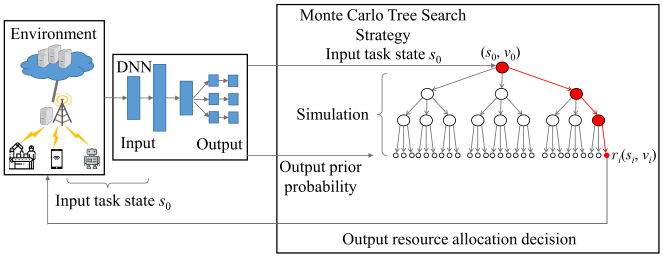


Fig. 14. An illustration of MCTS algorithm in DRL.

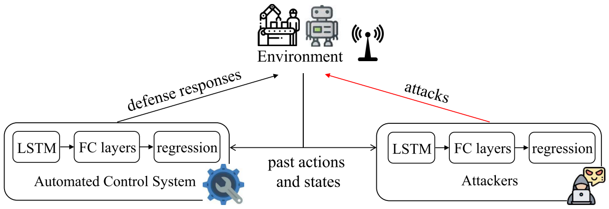


Fig. 15. A MARL-based attack-defense model for cyber-physical security.

can further investigate the application of prediction neural networks on facilitating the deployment of DRL algorithms in large-scale IIoT systems.

2) Resource Trading: Although computation offloading is considered as a promising technology to support IIoT, the high resource-rental price caused by the current inflexible pricing models is the major barrier to its wide adoption. To develop a mutually beneficial and demand-driven pricing model for resource trading, the authors in [118] applied DRL to the pricing problem between cloud providers and resource renters, which was formulated as a two-stage Stackelberg game. In this model, the cloud provider who acts as the leader first sets a price, then the renters follow and determine the resources to be purchased. To achieve the Nash equilibrium of the two-stage Stackelberg game, the authors proposed a multi-agent Q-learning-based algorithm called WoLF-PHC. At the beginning of each decision epoch, the Q-learning agent deployed in the cloud provider sets the price according to the earlier demands of renters. Then, the Q-learning agents deployed in the resource renters determine their service demands. Through interactions and policy iterations, the DRL agents deployed on both sides converge to stationary policies. Simulation results demonstrate the advantage of the proposed WoLF-PHC algorithm over the traditional Q-learning algorithm in terms of convergence performance and resource utilization. Nonetheless, the above work leaves much room for improvement, such as the prediction of the workload traces of renters.

C. Preventive Maintenance and Cyber-Physical Security

The wide adoption of IoT devices in the industry puts forward higher requirements for system stability and security. The lack of effective countermeasures is the potential bottleneck for future IIoT applications. Fortunately, with the development of AI, DRL has shown great performance in solving complex problems without prior knowledge of the system.

1) Preventive Maintenance: Firstly, we focus on preventive maintenance, which refers to reducing the likelihood of equipment failure by regular equipment maintenance, such as machine replacement and calibration. Motivated by this requirement, the authors in [119] presented a pioneering work that uses Q-learning for preventive equipment replacement. In general, equipment status in industrial systems deteriorates with time, which consequently results in equipment failures. It is a dilemma that the cost of high-frequency equipment replacements may outweigh the economic benefits, while inadequate replacements may increase the failure likelihood. As a result, the objective of the proposed algorithm is to determine the ideal replacement timing. In the model, the state is represented by the deterioration of equipment quality. Compared with two heuristic strategies, the proposed algorithm is prone to the least replacement times without jeopardizing the operational stability in the simulation.

The replacement timing problem of industrial equipment can also been found in [120] and [121]. Different from [119], the authors in [120] employed the Monte-Carlo RL algorithm [122] to scale the traditional DRL approaches to problems with large search spaces, which directly learns from the average sampling returns of simulated experiences. In addition, Wang *et al.* [121] considered the preventive maintenance problem in the context of POMDP. To deal with the partial observability, a fully distributed multi-agent deep Q-learning algorithm was proposed. In this case, the reward of each independent agent is set as the negative value of the overall system cost. Extensive experiments are conducted to demonstrate the effectiveness of the proposed algorithm.

2) Cyber-Physical Security: Secondly, we focus on the cyber-physical security, which refers to technologies protecting the network, data, devices, and control systems from attacks or unauthorized access. The dependence of industrial automation systems on data collection and sharing from IoT devices exposes their vulnerability to cyber-physical attacks. Therefore, the authors in [123] proposed a multi-agent deep Q-learning-based algorithm and formulated the problem as a noncooperative game. As illustrated in Fig. 15, the attackers attempt to manipulate the transmitted data via false data injection and smart jamming so as to induce equipment collisions. On the other hand, the automated system aims to coordinate the interdependent equipment so as to ensure the robustness. Considering the unobservability of the system, DRL agents are deployed to independently control the automated system and attackers.

Furthermore, the authors in [123] introduced LSTM to extract system features from past actions and states. The core idea of the LSTM is store and control the information flow with LSTM units, including input gates, output gates, and forget gates. Let d_t and y_t denote the input vector and output vector to the LSTM unit, respectively, where the subscript t denotes the time step. The input gate, output gate, and forget gate are in the following form:

$$\begin{aligned} i_t &= \sigma(W_i d_t + U_i y_{t-1} + b_i), \\ o_t &= \sigma(W_o d_t + U_o y_{t-1} + b_o), \\ i_f &= \sigma\left(W_f d_t + U_f y_{t-1} + b_f\right), \end{aligned} \tag{24}$$

where W_i, W_o, W_f and U_i, U_o, U_f denote the weight matrices, b_i, b_o and b_f denote the bias vectors, and σ denotes the sigmoid function. During the training process of DRL, the LSTM uses the selected transitions to update the weights and bias of the LSTM units and consequently learns to identify long-term dependencies by controlling the information flow. Simulation results show that the proposed algorithm can converge to a mixed-strategy Nash equilibrium and improve the robustness against cyber-physical attacks. A similar work is presented in [124], where a deep *Q*-learning agent works as an effective tool to identify malicious attacks on data transmission. However, because many security problems do not have predefined metrics, most recent studies resort to some indirect evaluations which may be inaccurate. In view of this gap, future investigation is expected to design effective evaluation metrics.

D. Automated Operational Control

The widespread deployment of IoT devices in the industry makes it possible to realize industrial automation, which provides automated operational control of manufacturing systems (such as robots) without human interaction. The objective of automated operational control is to maximize economic benefits and ensure production safety. In the following, we review the DRL applications on automated operational control.

1) Industrial Robot Control: A representative example of IIoT automation is robot systems, which can be used to perform various tasks, such as assembly, disassembly, picking, and insertion. Meyes *et al.* [125] proposed an deep *Q*-learning-based algorithm to achieve high precision in robotic assembly, which requires no extensive expertise of the physics model. In practice, the proposed algorithm takes the raw input data that is collected from force and position sensors, and adopts multiple LSTM layers to capture the hidden information from raw inputs. Simulation results show the effectiveness of the proposed algorithm under various robotic systems. However, such a trained DRL-based agent relies strongly on the precision of sensor input. As an extension, Schoettler *et al.* [126] considered a more practical scenario where sensor data may be imprecise or incomplete due to measurement errors. Specifically, they introduced artificial goal perturbations during the training process to mitigate the sensor noise. The robustness of the proposed algorithm is verified in the experiments. Moreover, instead of focusing on single robot control, the authors in [127] and [128] considered the problem of cooperative multi-robot control and proposed a MARL-based algorithm. Experimental results show that the proposed algorithm can better coordinate the operations of multiple robot systems to deliver seamless control.

2) Manufacturing Dispatching: Manufacturing dispatching refers to determining the operation sequence of a series of tasks, which is strongly desired in industrial automation. However, traditional solutions such as heuristics algorithms fail to respond in a timely manner for large-scale systems. Motivated by this, Zheng *et al.* [129] studied the potential of DRL on manufacturing dispatching. The two main contributions of this work are as follows. First, the state is represented by a two-dimensional matrix, which includes task characteristics, task queue states, and machine states. Second, the authors enhance the proposed algorithm with transfer learning in regard to scalability and generalization. Experiment results show the effectiveness and broader generalization ability of the proposed algorithm. With one step further, Mao *et al.* in [130] considered dispatching tasks with multiple resource demands and proposed a novel policy gradient method that promotes the monotonic improvement of policy iteration. Specifically, the advantage-value function $A(s_t, a_t)$ is introduced, which calculates the advantage value with a single function approximator, namely, $V(s_t)$. Using temporal-difference as an estimator, the advantage values can be calculated as follows.

$$\begin{aligned} A(s_t, a_t) &= Q(s_t, a_t) - V(s_t) \\ &= R(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t), \end{aligned} \tag{25}$$

where $Q(\cdot,\cdot)$, $V(\cdot)$, and $R(\cdot,\cdot)$ are defined in Section II. Accordingly, the update rule in Equation (15) can be rewritten as:

$$\theta \leftarrow \theta + \alpha_\theta A(s_t, a_t) \nabla_\theta \ln \pi_\theta(a_t|s_t). \tag{26}$$

Simulation results show that the introduction of advantage function contributes to stable gradient estimation. Even so, DRL is far from being widely applied in real-world scenarios. One major obstacle is the huge cost of trial-and-error. For example, a reckless action generated from a poorly designed exploration policy may cause life-threatening situations in industrial operation. This technical limitation encourages future work in designing a more intelligent exploration policy.

Summary: In this section, we have reviewed the DRL applications on IIoT systems. Table VI presents a summary of the related studies and main approaches adopted. It is encouraging that DRL algorithms can provide effective solutions to improve the performance of IIoT systems. Notably, the future IIoT system is envisioned to be associated with a large number of IIoT devices and a massive volume of data, thereby leading to higher scalability requirement. This issue is worthy of in-depth investigation. DRL methods such as HRL may be an ideal solution for high-level control of massive IIoT devices.

VI. DRL Applications on IoT-Enabled Mobile Crowdsensing

MCS refers to a technology where a group of individuals leverage their mobile devices equipped with sensors to collect, share and process data. With the proliferation of IoT devices and mobile sensing techniques, IoT-enabled MCS is emerging as an effective paradigm that enables a larger sensing area with a lower cost. Along with its benefits, however, IoT-enabled MCS faces a series of challenges, such as the quality of sensory data, incentive mechanism, security and privacy. In view of the potential applicability of DRL, we review the DRL applications on IoT-enabled MCS in this section.

A. Sensing and Communication

Firstly, we discuss the DRL applications on IoT-enabled MCS from the sensing and communication layer. Compared

TABLE VI
DRL APPLICATIONS ON INDUSTRIAL IOT

Algorithms	Applications	4Cs problems	MDP structure			Refs.
			State	Action	Reward	
Deep Q-learning	energy management	- control - communication	- on-off status - communication modes - cache status, etc.	- power control - channel selection	- energy consumption	[106] – [108]
	QoS network control	- communication - caching	- user status - available resources, etc.	- power control - channel selection	- network throughput	[109] – [111]
	computation offloading strategy	- computing - communication	- server capacities - task information - network states, etc.	- offloading decision	- task completion reward - execution cost	[113] – [117]
	preventive maintenance	- control	- deterioration status	- replacement timing	- system stability - replacement cost	[119]
	malicious attacks detection	- control - communication	- network status - user reputations	- malicious users	- network stability	[124]
	robotic assembly	- control	- sensor raw inputs	- robot motions	- completion rate	[125], [126]
	manufacturing dispatching	- control	- task characteristic - machine states - task queue states	- task sequence - allocated resources	- manufacturing profit	[129]
Multi-agent deep Q-learning	pricing models	- control	- server capacities - user workload (all) - user bidding (all)	server: - rental price	- net profit	[118]
			- server capacities - workload (observable) - rental price	user: - offloading decision	- task cost	
	preventive maintenance	- control	- deterioration status	- replacement timing	- system stability - replacement cost	[121]
	anti-jamming	- control - communication	- network status	- defense response	- network stability	[123]
				- jamming attacks	- jamming affect	
	robot control	- control	- sensor raw inputs	- robot motions	- completion rate	[127], [128]
Policy-based DRL	manufacturing dispatching	- control	- task characteristics - machine states, etc.	- task sequence - allocated resources	- manufacturing profit	[130]

with traditional static sensor networks, IoT-enabled MCS not only provides a more flexible solution, but also presents greater challenges to sensor control and communication services.

1) Adaptive Sensor Control: Considering the limited battery capacity of mobile devices, continuous passive sensing of equipped sensors may drain the battery rapidly. To extend the lifetime of mobile devices, Chen *et al.* [131] proposed an adaptive sensor control algorithm based on *Q*-learning. This work considers an MCS applications for area coverage, and the task of each MCS participant is to adjust the device operation model (active or sleep) to improve energy-efficiency while achieving a satisfying area coverage ratio. In particular, the input state of *Q*-learning refers to the states of all devices, including position, sensing ranges, sensing power, energy preservation, etc. For comparison, the random and LEACH [132] algorithms are used as baselines. Simulation results show that the proposed algorithm can effectively extend the lifetime of IoT-enabled MCS systems while maintaining a higher coverage ratio. In contrast to [131], the authors in [133] and [134] take the independence of MCS participants into account. In such scenarios, it is infeasible to implement a central control agent. Alternatively, the authors proposed a multi-agent *Q*-learning-based algorithm, which can balance the sensing costs and rewards, and reach the Nash equilibrium. However, for the accomplishment of the above approaches, the system states much be accurately evaluated, which is not practical for real-life scenarios. Therefore, the future work might consider the potential evaluation errors of system states, possibly due to sensor failure or packet loss.

2) QoS Provisioning for High-Mobility Network: The high flexibility of IoT-enabled MCS also brings the challenge in the QoS provisioning and makes the traditional approaches inapplicable, since they are mostly designed for static sensors networks. Fortunately, the data-driven DRL offers an attractive solution to learn the mobility patterns of MCS participants. For example, Xiao *et al.* [81] focused on the multi-UAV sensing problem and proposed a MARL-based algorithm to learn the mobility patterns of UAV and improve QoS. In the considered IoT-enabled MCS model, there exist smart jammers that interfere or disturb the services, hence the interactions between UAVs and smart jammers are formulated as a mixed-strategy game. Each UAV works as an MCS participant and adaptively selects its relay nodes based on the observed states, while the smart jammers adjust its jamming power based on relay node selections of UAVs and the network state. Without knowing the attackers beforehand, the MARL is introduced for each UAV to avoid jamming. Simulation results reveal that the MARL-based algorithm can achieve a much higher UAV utility and a lower BER than the traditional *Q*-learning scheme. Besides, DRL is also broadly applied in many other IoT-enabled MCS scenarios to address the QoS provisioning challenges brought by high mobility, such as routing protocol [135], [136], connectivity preservation [137], and network security [79], [82]. However, the performance evaluations of the above studies are mostly conducted in the simulation. We highly expect that future studies include performance evaluation on real-world experiments.

B. Dynamic Participant Recruitment

Participant recruitment refers to the technique that encourages a set of mobile users with IoT sensing devices to accomplish the data collecting, sharing and processing tasks. In general, the deployment and maintenance of traditional static sensor networks are economically expensive. In the following, we review the DRL applications on developing adaptive recruitment strategies, which are classified into platform-centric and user-centric according to their implementations.

1) Platform-Centric Participant Recruitment: Firstly, we discuss the platform-centric participant recruitment, which is the most common type. In such a scheme, a central decision-making agent is implemented to directly control all the MCS participants and maximize the overall utility of IoT-enabled MCS systems. In [138], the authors presented the first attempt that applies DRL to the participant recruitment in the smart city. To successfully accomplish the MCS tasks and recruit high-quality workers, a DRL-based context-aware recruitment strategy was developed. The QoS of a worker is evaluated by its extrinsic and intrinsic abilities and its recruit costs. Then, they used DRL to update the evaluation parameters of candidate participants based on the status of MCS task accomplishments. Numerous experiments were conducted based on simulations and MIT Reality data [139], where the proposed algorithm achieves the highest long-term utility of the MCS system and social welfare.

The participant recruitment model can also be found in [140] and [141]. In particular, the authors in [140] took the transportation system as a case study and developed a robust IoT-enabled MCS framework that supports active participant recruitment, data validation, and local data processing. Firstly, they carried out the participant recruitment based on the reverse Vickrey-Clarke-Groves auction algorithm [142] for monetary reward determination. Then, each MCS participant was set to collect data, including traffic and road conditions, driving behavior, vehicle velocity, etc. Lastly, data validation was taken before delivering the data to the sensing platform, which effectively reduces the inefficient sensory data and maximizes the sensing profit. Simulation results demonstrate the robustness and efficiency of the proposed algorithm. Furthermore, to improve the scalability and accelerate the learning process, Zhan *et al.* [141] combined game theory with DRL. Simulation results show that this improvement enables DRL agents to converge to a stable state at a faster rate.

2) User-Centric Participant Recruitment: Next, we discuss the user-centric participant recruitment, where the behaviors of each MCS participant is determined by its own will rather than by a central agent. Instead of maximizing the global utility of IoT-enabled MCS systems, the user-centric scheme focuses on achieving a Nash equilibrium. For example, the authors in [143] studied the incentive mechanism and participant recruitment strategy from the perspective of mobile users. To elaborate on this scheme, they considered an IoT-enabled MCS model consisting of a service provider and multiple mobile users. The former is to publish the sensing tasks with a time-dependent reward budget, and the latter would perform sensing tasks with their equipped sensors. To characterize MCS participants, the authors formulated the system as a multi-agent MDP in which a user is needed to decide its effort level with no observation of other user decisions. To achieve an equilibrium, an intelligent online sensing scheme called IntelligentCrowd was proposed with the help of the multi-agent actor-critic algorithm. Simulation results show that the IntelligentCrowd can efficiently learn the optimal policy under various environmental settings. In addition, Zhan *et al.* [144] extended the single-leader multi-follower Stackelberg game in [143] to multi-leader multi-follower game, where mobile users receive requests from multiple service providers. In the future, studies on applying the MARL technique in real-time MCS scenarios are expected.

C. Data Collection and Processing

The accomplishment of an MCS task requires decision making that typically involves data collection (e.g., designing the required sensors and their sensing targets) and data processing (e.g., designing whether to perform the tasks locally or offload to the cloud/edge servers). With the limited computational power and energy supply of mobile devices, it is essential to develop an optimal strategy for data collection and processing.

1) Energy-Efficient Data Collection: In general, mobile devices are constrained with lifetime and sensing range because of the limited battery capacity. As a result, it is crucial to design an energy-efficient data collection strategy for IoT-enabled MCS systems. The authors in [104] proposed to use the multi-agent DDPG algorithm to direct the sensing and motion of UAV and autonomous vehicle. Similarly, the authors in [145] and [146] discussed the energy-efficient data collection in the UAV context. Different from [104], they considered a more complex scenario where UAVs are equipped with wireless charging devices. To collect the highest priority data and improve energy efficiency, they utilized the multi-agent DDPG algorithm for the cruise route control and energy charging. In addition, they considered that the traffic flow to ensure a UAV can return the origin position with enough energy. Simulation tests based on real datasets have proven the robustness and effectiveness of the proposed scheme in terms of training rate, energy efficiency, geographic fairness, and data collection ratio.

2) Data Inference for Data Collection: Data inference refers to the process of deducing the data quality in the unknown area based on historical experience and background knowledge. The abundant of available data makes it possible to infer sensing context and user behavior. For instance, the authors in [147] studied a sparse IoT-enabled MCS model for large-scale urban sensing, where the data collection problem was formulated as a cell selection problem, as illustrated in Fig. 16. In the model, the state is represented by a matrix that contains the data collection conditions in the sensing cycle. The action is modelled as the selected possibility of cells. And the reward is defined as the difference between sensing rewards

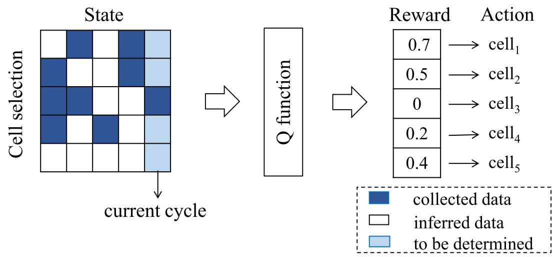


Fig. 16. The system model of DR-Cell.

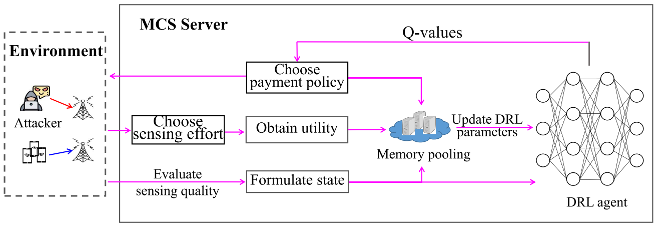


Fig. 17. The deep Q-learning-based MCS security payment strategy.

for satisfying the required inference quality and the cost of data collection. For data inference, a deep Q-learning-based cell selection strategy, termed DR-Cell, was proposed to guarantee high inference quality. Moreover, they considered a reducing of training data via transfer learning. Simulation results show that the DR-Cell algorithm can reduce the sensed cells up to 15% without loss of data inference quality compared to the QBC algorithm [148]. In [149], the DR-Cell algorithm was further enhanced by using the QBC algorithm in the initial training process, which greatly reduces the cost of trial-and-error. However, their work cannot provide an effective online mode with cell selection, which remains to be solved.

3) Fog Computing for Data Processing: In most IoT-enabled MCS applications, mobile IoT devices require to perform data processing tasks after collecting data, which may be computationally intensive. Li *et al.* [150] presented an innovative work that uses deep Q-learning for the task scheduling of fog computing to address the challenges. The system state is the information of all fog nodes and task requests. The reward is a composite of three components: (i) the size of uploaded data, (ii) the bandwidth cost, (iii) the energy consumption. By adaptively scheduling tasks according to the system states, the well-trained deep Q-learning agent can maximize the MCS utility with the minimum bandwidth cost and energy consumption. Remarkably, the data collection strategies determine the locations of mobile devices in the IoT-enabled MCS context and thus determine the available fog nodes near the mobile devices. Therefore, data collection and processing can be jointly considered for better performance in future work.

D. Crowdsensing Participant Privacy

In an IoT-enabled MCS system, the sensing data may carry sensitive personal information, such as the real-time location of crowdsensing participants. This privacy concern may frustrate mobile users to participate in the sensing activities and the development of MCS. We next review the applications of DRL concerning the above issues.

1) Secure and Privacy-Preserving Communication: Firstly, we discuss the privacy issues from the perspective of communication security. Due to the fragmented nature and the ineffective security mechanisms of most IoT-enabled MCS systems, malicious attackers may access the sensitive information of MCS participants and provoke data leakage. To tackle these challenges, Shakeel *et al.* [124] proposed a deep Q-learning-based algorithm that examines the potential threats. The state is specified by the extracted network features, such as the channel states and received signal strength, as well as the channel impulse response that contributes to evaluate the network security. The objective is to analyze the arrival packets and identify the malware affected data. Simulation results show the deep Q-learning framework is superior to three popular machine learning techniques in terms of detection accuracy achieved by Multi-layer Perceptron, Back Propagation Neural Network, and Learning Vector Quantization. Furthermore, in [151] and [152], DRL is also used to enhance communication reliability and detect cyber-physical attacks. We observe that the above approaches are designed against a single known attack. In future work, it can be considered to design a DRL algorithm that can adjust its parameters online to accommodate and detect multiple known/unknown attacks.

2) Tradeoff Between Data Quality and Participant Privacy: Next, we discuss the participant privacy issues from another perspective: selfish MCS participants may submit faked sensing data to servers out of privacy concerns. Accordingly, to balance the tradeoff between data trustworthiness and user privacy, DRL can be applied to IoT-enabled MCS systems, i.e., [153], [154], [155] and [141]. The authors in [153] discussed the deployment of data validation agents on the edge of networks, which can ensure data quality and protect user privacy. The validation of sensing data can also be used to improve the efficiency of IoT-enabled MCS systems by preventing poor-quality or useless data from taking up network resources. In [154], the authors proposed a deep Q-learning-based MCS payment strategy. As illustrated in Fig. 17, the interactions between multiple MCS participants and an MCS server is formulated as a one-leader multiple-follower Stackelberg game. In such a game, the MCS server is first to broadcast its payment policy to MCS participants according to the evaluation of their sensing qualities. Then, each participant is to choose a sensing effort (e.g., time and power) and receive the payment accordingly. Compared to non-DL methods, the proposed deep Q-learning-based algorithm shows higher sensing quality and utility with a lower faked sensing rate. It outperforms the Q-learning based algorithm by 96.0% and 31.5% in the utility of MCS servers and the sensing quality, respectively, after 200 time-slots.

Unlike existing studies, the authors in [155] discussed privacy protection under a dynamic and time-varying IoT-enabled MCS scenario. In this work, each participant uses a time-variant and self-specified privacy-preserving level to perturb its sensing data. To handle the demand uncertainties and make optimal pricing for sensing tasks while protecting participant's privacy, the authors considered a semi-trustworthy MCS platform and proposed a Q-learning-based algorithm to determine

TABLE VII
DRL APPLICATIONS ON IOT-ENABLED MOBILE CROWDSENSING

Algorithms	Applications	4Cs problems	MDP structure			Refs.
			State	Action	Reward	
(Deep) Q-learning	extending sensor lifetime	- control - communication	- task information - energy reserve	- mode switch (sleep/active)	- long-term task rewards	[131]
	QoS network control	- communication - control - caching	- network status - device status	- relay node selection	- QoS evaluation	[82], [136]
	participant recruitment	- control	- participant status - price and bidding	- worker selection	- completion rate - net profit, etc.	[138] – [141]
	maximizing data collection rate	- communication - caching	- participant status - network status	- the sensing and motion of device	- data collection rate	[147], [149]
	minimizing data processing cost	- computing - communication - caching	- task information - server capacities - device status	- offloading decision	- processing cost	[150]
	security and privacy management	- control	- network status - user privacy level	- identify attacks	- network security - user privacy	[124], [151], [152]
	balance between privacy and data quality	- control	- data quality - user status, etc.	- data validation - payment policy, etc.	- data quality - user privacy	[141] [153] – [155]
Multi-agent deep Q-learning	balance sensing costs and rewards	- control - communication	- task information - sensing cost	- sensing effort	- sensing cost - sensing reward	[133], [134]
	anti-jamming	- control - communication	- radio channel state	- relay node selection	- UAV utility	[81]
				- jamming attack	- jamming affect	
	QoS control in network	- communication - control - caching	- transmission BER - network status - device status	- channel selection, - relay node selection, etc.	- QoS evaluation	[79], [135]
	participant recruitment	- control	- participant status - task information - price and bidding	server: - recruit prices	- net profit	[134], [143], [144]
				participant: - bidding	- task completion rates and cost	
DPG/DDPG	network connectivity control	- communication - control	- robot status	- robot velocity	- connection rate	[137]
	data collection strategy	- communication - control	- task information - device status	- the sensing and motion of device	- energy consumption	[104], [145], [146]

the pricing policy. Specifically, they modeled the IoT-enabled MCS system as an MDP, in which the state was related to the current levels of privacy-preserving for all participants. Then, the Q-learning-based algorithm was used to optimize the system utility according to the perceived reward and the observed system state. In this way, the pricing policy could be adjusted dynamically, so that the different levels of privacy protection for different users can be satisfied.

Summary: In this section, we have reviewed how DRL can be applied to address the challenges of IoT-enabled MCS. A summary is presented in Table VII. Although the above studies have achieved encouraging results, we note that there are still a variety of challenges in the development of IoT-enabled MCS technology. Meanwhile, both the efficiency and stability of algorithms need to be further improved. In summary, these challenges require to be examined in detail in future work.

VII. DRL APPLICATIONS ON BLOCKCHAIN-EMPOWERED IOT

Blockchain, as an emerging technology, has been gaining popularity in many areas. Owning to its decentralization, transparency, immutability, and security, blockchain has been considered as an effective tool to bestow data trust, security and system reliability on traditional IoT systems [156]. However, a blockchain-empowered IoT system often demands huge computational resources and suffers from low transaction throughout. In light of this challenge, we then discuss the applications of DRL on blockchain-empowered IoT as follows.

A. 5G-Enabled Blockchain Networks

One of the major problems of the blockchain-empowered IoT is the scalability of IoT networks. Compared with 4G, the 5G communication networks can bring higher capacity, lower latency, much higher transmission bandwidth, massive number of accesses. In the following, we discuss how DRL can be applied to release the true potential of 5G technology.

1) SDN Control Plane Synchronization: Considering the sophisticated design and heterogeneous architecture of 5G networks, a flexible control plane is required whereas traditional hardware-based approaches cannot provide. Recently, SDN has emerged as a promising network architecture for optimal control, where each independent SDN controller is in charge of a network domain and can synchronize with each other. However, due to the variability and uncertainty of network environments, the design and synchronization of the control plane for 5G SDN is a great challenge. Inspired by this issue, the authors in [157] leveraged blockchain and DRL to design a robust SDN control plane for 5G networks. On the one hand, blockchain is utilized to protect the sensitive control information in distributed controllers. On the other hand, DRL is utilized to overcome the design complexity of the SDN

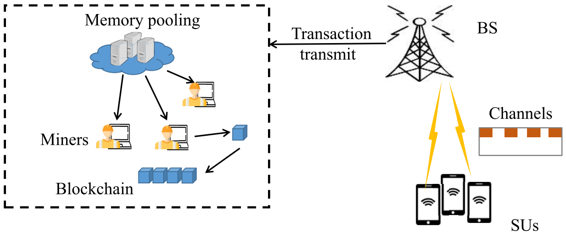


Fig. 18. A cognitive radio based blockchain network.

control plane. The synchronization problem is formulated as an MDP, where the states and rewards are collected using the SDN interface. Benefit from the abundance of available data in SDN, the data-driven DRL is appealing to develop the optimal synchronization policy. However, the authors in [157] did not provide a specific implementation of the proposed scheme, so that quantitative performance evaluations are missing. Besides, the integrity of sensitive information with DRL should also be considered in the design of the SDN control plane.

2) Trusted Routing Scheme: To cope with the ever-increasing data traffic and security risks, the trusted routing scheme is of great significance in future 5G networks. To identify malicious nodes, the authors in [158] put forward a trusted routing scheme based on the blockchain and RL. In the work, a threat model was first given. Then, a blockchain-based network architecture was designed, where the digitized information of routing packets is recorded in the smart contract. In this regard, the routing nodes can obtain the trusted routing information that they need from the blockchain network instead of their neighboring nodes. Moreover, in order to help a routing node select the most reliable and effective path, a named RLBC routing algorithm was designed by using RL. Because the routing information recorded in the blockchain network is traceable and tamper-proof, the RLBC algorithm can train its model with reliable data. Compared with other related algorithms in [159], [160], and [161], the RLBC algorithm shows lower latency and higher throughout even in a routing environment with 50% malicious nodes. Notably, the authors intend to introduce the RL into the smart contracts in future work, which is challenging since incumbent smart contracts in the blockchain are not suitable to perform complex operations.

3) Optimizing Transaction Transmission Policy: The blockchain-empowered IoT system is regarded as a promising paradigm to manage data in a decentralized but dependable manner. The transactions, such as the sensing data of IoT devices, can be recorded and validated in the blockchain. In such a system, it is essential to optimize transaction transmission policy to mitigate transmission failures. In view of this, the authors in [162] studied the combination of blockchain and CRN, in which an IoT device can transmit the sensing data to the blockchain network via the idle channels of primary users. Fig. 18 shows the system model. To enable each secondary user (SU) to make the optimal decision (i.e., channel selection and transmit decision), a double deep Q-learning-based algorithm was utilized to maximize the number of successful transaction transmissions. The action is the selected channels of SUs to transmit transactions. The state is related to both channel state (idle or busy) and mempool state. The reward function is determined by the positive utility, channel access, and the transaction fee. In addition, to address the overestimation problem caused by the max operator in Equation (9), double deep Q-learning implements an additional DQN called target DQN. Let $Q_{\omega}(s_t, a_t)$ and $Q_{\omega'}(s_t, a_t)$ denote the original DQN and target DQN, respectively. In double deep Q-learning, $Q_{\omega}(s_t, a_t)$ and $Q_{\omega'}(s_t, a_t)$ are simultaneously used to calculate the action-value, which can be represented by the following:

$$\begin{aligned} Q_{\omega}(s_t, a_t) \leftarrow{}& Q_{\omega}(s_t, a_t) \\ &+ \alpha\Bigg(r_{t+1} + \gamma Q_{\omega'}\left(s_{t+1}, \arg\max_{a \in \mathbb{A}} Q_{\omega}(s_{t+1}, a)\right) \\ &- Q_{\omega}(s_t, a_t)\Bigg), \end{aligned} \tag{27}$$

where the value of selected action is fairly evaluated with target DQN, which periodically updates parameters at a slower pace. Simulation results show that the proposed double deep Q-learning-based algorithm gains reward more than 41% compared with the traditional Q-learning.

B. Resource Allocation for Blockchain Mining

Blockchain is typically associated with computationally-intensive blockchain mining tasks, which refer to the process of transaction being validated and a new block being appended to the blockchain. In particular, IoT devices generally have intrinsic restraints on computational resources and storage capacity. This limitation motives the system designers to leverage the computing resources for the cloud/edge servers.

1) Transaction Caching: To handle the unbalance of the rapidly growing blockchain size and the limited storage capacity, transaction caching is proposed by allowing resource-constrained IoT devices to request transactions from local caching providers. The authors in [163] built a secure content caching environment based on consortium blockchain and studied the optimal transaction caching strategy. To achieve maximum system utility, they utilized the DDPG algorithm to develop a novel content caching scheme in the blockchain-empowered IoT system. Specifically, the state is made of caching content states, the number of available caching resources and the bandwidth provided by a local caching provider. The action is made of the amount of bandwidth and a set of binary values that represents the request decision. And the reward function is determined by the system utility. Simulation results show that the proposed scheme achieves a higher system utility than the benchmark caching scheme. This work provides a groundbreaking perspective on applying DRL to transaction caching. For future work, the applications of other state-of-the-art DRL algorithms to the transaction caching can be further investigated.

2) Computation Offloading for Blockchain Mining: As mentioned above, the computation offloading can provide additional resources for IoT devices. However, the high complexity and uncertainty of the blockchain network pose great challenges to the traditional offloading strategies. The authors in [164] studied a novel blockchain-empowered IoT architecture in which the miners acting as mining agents offload blockchain mining tasks to cloud servers and manage network resources dynamically. Then, a computation offloading algorithm based on dueling deep Q-learning was proposed to solve the joint optimization problems of user access selection, computational resource allocation, and network resource allocation. Note that the above work only considers the blockchain mining tasks. However, many data processing tasks such as object tracking are also conducted in IoT devices, which may compete with mining tasks for computational and network resources. As an extended work, the authors in [165] studied the computation offloading problem of both blockchain mining tasks and data processing tasks. In particular, prioritized experience replay is introduced for efficient learning, where the transitions are labeled with different weights and thereby important transitions are selected as samples more frequently. Formally, the sampling probability of a transition i can be calculated by:

$$P(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}}, \tag{28}$$

where α denotes how much prioritization is used in the replay buffer. In addition, to prevent bias from causing uncontrollable update distributions, prioritized experience replay adopts the importance-sampling weights as follows:

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^{\beta}, \tag{29}$$

where N is the size of replay buffer, and the exponent β starts from zero and is linearly annealed towards 1 throughout the learning process. Simulation results show that the proposed algorithm can maximize the total utility of the system and balance the conflicts of both tasks.

In addition, the computation offloading problem in blockchain-empowered IoT can also be found in [166] and [167]. Specifically, a double deep Q-learning-based algorithm was deployed in user equipment, cloud servers, and edge servers to derive the optimal strategy. Different from [164] and [165], the authors integrated DRL with federated learning, in which the devices and edge nodes can collaborate to exchange learning parameters to obtain a better model inference and accelerate the training process without violating user privacy. Nonetheless, there are still some problems that deserve further study. For example, both the slow convergence of DRL training and the high cost of trial-and-error prevent DRL from becoming a generic solution for blockchain-empowered IoT systems. Therefore, different types of services shall be provided with differentiated supports. Moreover, a fine-grained collaboration among IoT devices and edge servers should also be considered.

C. Large-Scale IoT Systems and Scalability

The integration of blockchain technique and IoT systems suffers from significant scalability issues, which is the major bottleneck to apply blockchain to large-scale scenarios. The recent advancements of DRL have aroused a great interest in applying DRL to improve the performance of blockchain systems in terms of scalability. A review of these pioneering studies is given as follows.

1) Improving Blockchain Performance: The performance of a blockchain system is determined by three properties: throughput, decentralization and security. Although current blockchain frameworks and consensus mechanisms (i.e., Proof of Work and Proof of Stake) show superior performance in the context of decentralization and security, they also exhibit low throughput and are unsuitable to handle massive data. The first attempt that leverages DRL in blockchain systems was presented in [168], where a well-known consensus mechanism called Practical Byzantine Fault Tolerant was considered. To improve the blockchain throughput, the authors adopted the deep Q-learning to adaptively select block producers and adjust block size as well as block interval. The state is represented by the average transaction size, stakes distribution, computing capability of all nodes, and the data transmission rate between each pair of nodes. The reward is determined by transaction throughput, decentralization constraints, and security constraints. Simulation results show that the proposed scheme enhances the system's overall throughput without violating the constraints of decentralization and security. Furthermore, the authors extend their previous work to two advanced BFT protocols: Zyzzyva [169] and Quorum [170]. Then, a thorough performance analysis was given in [171]. However, there is still room for improvement. For example, a DRL-based adaptive scheme can be considered in the consensus process of blockchain-empowered IoT systems.

2) Efficient Data Collection: Next, we discuss the scalability issues of blockchain-empowered IoT systems from another perspective – the communication overhead caused by transaction transmission. Data collection and sharing are becoming more and more common in IoT applications. Although the introduction of blockchain can ensure the security in data sharing among IoT devices, the maintenance of blockchain may require the competition for network resources. In this context, the authors in [172] integrated Ethereum and DRL to develop a secure and efficient scheme for data collection and sharing. They proposed a distributed MARL solution that deploys an independent DRL agent at each IoT device. As shown in Fig. 19, each DRL agent first obtains an observation that includes the data distribution, the locations of IoT devices and the past trajectories. Then, based on the observation, it takes actions involving the moving direction and the distance. Lastly, it receives a reward from the environment, which is constituted by data collection amount, energy consumption and geographical fairness. Simulation results demonstrate that the proposed DRL-based distributed scheme increases data collection ratio by 34.5% compared with the random algorithm. However, it shows greater energy consumption compared with the random algorithm due to its high costs for data collection. Moreover, an IoT device is expected to perform multiple tasks simultaneously and involve multiple objectives in large-scale IoT systems. This issues motivates us to develop a DRL model with multi-objective optimization in the future.

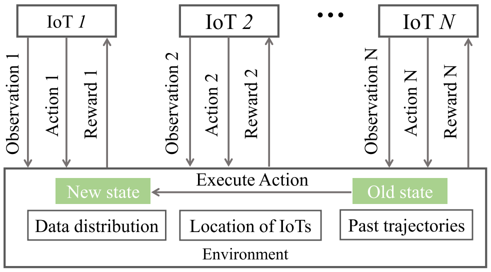


Fig. 19. An efficient data collection scheme for blockchain-empowered IoT.

D. Security and Privacy Protection

The blockchain is introduced into various IoT systems because of its inherent security properties. However, there is a lack of intelligent strategies to completely unleash the power of blockchain technology. To fill this gap, some studies have explored the applications of DRL on security and privacy protection. An overview of these studies is given as follows.

1) Access Control and Authentication: Access control and authentication are deemed as the key techniques to ensure security and privacy in the context of IoT systems. The authors in [173] studied the access control issues and proposed a dynamic and fully distributed security strategy. On the one hand, they utilized the blockchain to build a basic framework to store and compute access policies. On the other hand, they suggested using the DRL algorithm to achieve dynamic and automatic management for security policies, which can adapt to environment dynamic and device variability. Their proposals may obtain the optimal access control rules via learning feedback, whereas the work of [173] does not give a case study. In addition, the inherent defects of blockchain, such as the high computation overhead and high latency, are not taken into account. The DRL techniques are found to be used in some authentication studies to improve the effectiveness. For example, the authors in [174] employed a Q-learning-based physical-layer spoofing detection algorithm to authenticate the radio channel information. Simulation results show that the Q-learning-based algorithm can outperform the benchmark scheme in terms of the receiver utility by 2.6%.

2) Communication Security and Privacy: Next, we discuss the security and privacy challenges in IoT communication. As presented in [175], the authors studied the potential malicious onboard units (OBUs) attacks in VANETs, including jamming, spoofing, Sybil attacks and so on. The authors proposed to utilize the blockchain techniques to secure the storage, distribution, and broadcasting of data, thereby guaranteeing tamper-resistance. In addition, a Q-learning-based strategy is given to select reliable relays for OBUs in VANETs. In this scheme, the environment state is related to the role, reputation, location, and velocity. It is worth mentioning that the Q-values are initialized with prior experience at the beginning. This manner accelerates the training process. Compared with the traditional approach, the proposed algorithm can improve the average reputation by 3.52 after 3000 transmissions, while the system utility and packet delivery ratio increase by 151.64 and 55% respectively, after 600 packets are transmitted. Another use case can be found in [176], where the authors combined blockchain and MEC to build a traceable and tamper-resistant model for smart grid networks; this model is able to detect inappropriate use behaviors in wireless communications, thereby reducing the influence of channel interference and avoiding malicious attacks. To achieve the optimal energy transaction transmission, they formulated an Instant Energy Transaction Optimization problem and proposed a RL-based algorithm. Simulation results show that the RL-based algorithm can ensure high energy-efficiency in transaction transmission while maintaining security and preserving privacy.

Summary: In this section, we have reviewed the DRL applications on the blockchain-empowered IoT systems. We present a summary in Table VIII. In general, blockchain and DRL can be combined together in IoT systems, where the blockchain can establish a decentralized and reliable system and DRL can be used to address the environment dynamic with the provision of optimal strategies for a variety of applications. However, the seamless integration of blockchain and DRL is still rare in literature so as to let alone practical applications in the real world. Therefore, it requires efforts from an interdisciplinary perspective in the future.

VIII. Challenges and Future Directions

A. DRL Training Problem

Most existing studies assume that training DRL agents can be conducted in the cloud in a centralized manner. However, in real-world scenarios, this leads to the following challenges: 1) The training data shall be transferred to centralized servers, which may arise privacy issues. IoT users usually prefer keeping their privacy-sensitive data on hand. 2) Transferring a large amount of training data increases the communication overhead and the burden of IoT network. On the other hand, training DRL agents at the IoT sides also brings new issues: 1) The training process of DRL agents is extremely energy-consuming for end devices; 2) The limited computation capacity of end devices would cause a long delay or failure of large-scale data training. Therefore, how to appropriately train DRL agents shall be investigated.

It is natural to bring edge computing into DRL training process in a distributed manner, which can overcome the resource limitation of end devices and mitigate the privacy concerns. Edge computing supports DRL training in various aspects, such as algorithm design, network architecture, performance optimization, etc. It is worthwhile to study the distributed deployment of DRL agents at the edge. First, hardware devices and software frameworks customized for the edge facilitate effective DRL training and implementation. Second, well-designed computation offloading frameworks can enable DRL agents in resource-limited end devices, leading to more pervasive and more fine-grained intelligence. Third, federated learning can be integrated into the training of distributed DRL agents [166], which makes it possible to gain knowledge from massive data while ensuring privacy.

B. MARL Control Problem

In many real-world scenarios, it is impractical to implement a centralized control agent due to its complexity,

TABLE VIII
DRL Applications on Blockchain-Empowered IoT

Algorithms	Applications	4Cs problems	MDP structure			Refs.
			State	Action	Reward	
(Deep) Q-learning	SDN control plane synchronization	- control - communication	- SDN network status	- synchronize control	- SDN performance	[157]
	routing	- communication	- packets location	- forwarding packets	- delivered packets	[158]
	offloading strategy for blockchain mining	- control - communication - computing	- server capability - network status - workload, etc.	- access selection - resources allocation	- miner utility	[164]
	optimizing blockchain performance	- control - communication - computing	- stakes distribution - computing capacity - date transmission rate - transaction size, etc.	- select block producer block size, and block interval	- decentralization - latency - security - scalability	[168], [171]
	access control and authentication	- control	- network status - access requests, etc.	- detect illegal access	- network security	[173], [177]
	network security management	- control - communication	- network status	- detect attacks	- estimated error	[175], [176]
Double deep Q-learning	transmission strategy optimization	- communication	- channel status - mempool status	- channel selection	- access cost - transaction fee	[162]
	offloading strategy for blockchain mining	- control - communication - computing	- server capability - network status - workload, etc.	- access selection - resources allocation	- miner utility	[166]
Multi-agent Q-learning	data collection optimization	- control - communication	- locations - sensing times, etc	- moving direction - moving distance	- fairness - energy cost	[172]
	access control and authentication	- control	- network status - access requests, etc.	defense system: - detect illegal access	- detect accuracy	[174]
				attacker: - illegal access	- network damage	
DDPG	transaction caching strategy	- caching - communication	- content status - available resource	- binary value - bandwidth allocated	- system utility	[163]
	offloading strategy	- computing - communication	- server capability - network status, etc.	- offloading decision	- system network profit	[165]

communication delays, and privacy concerns. Accordingly, some recent studies have turned to MARL for a help, in which DRL agents are distributed across IoT devices. However, many challenges remain. First, the interactions between distributed DRL agents complicate the problem. This is because the lack of effective coordination mechanisms among distributed agents deteriorates the environmental stability and makes it difficult to converge. Second, knowledge sharing among different DRL agents is a common technology to accelerate the training process, but it also results in additional communication overhead in the IoT network. In this context, there is a tradeoff between communication efficiency and high performance.

Recent years have witnessed an increase in MARL-related research, providing some critical open directions that have yet to be explored. For instance, a state-of-the-art MARL algorithm called common knowledge reinforcement learning was proposed in [178], which leverages common knowledge among different agents to achieve complex decentralized coordination. However, this work has a limitation of failing to consider the communication delay for reaching consensus on common knowledge. Therefore, the tradeoff between communication efficiency and high performance in MARL needs to be further studied in the future. In addition, a well-designed MAS framework facilitates MARL collaboration and knowledge management, which is worthy of in-depth investigation.

C. Reducing the Gap Between Simulation Environment and Real-World Scenario

The training process of DRL agents requires massive data to achieve satisfactory performance. For many IoT scenarios, training data is generally inaccessible in their early development, thus there exist few referential datasets. In practice, DRL agents are trained not from scratch in real-world scenarios, but pre-trained in a simulated environment before deployment, which further raises the importance of reducing the gap between simulations and real-world systems. On the one hand, most IoT systems have delay in device sensing or reward feedback. This is because the actions may need to be processed (e.g., safety check) before execution or it may take time for the actions to take effect. On the other hand, many real-world IoT systems are partially observable. For example, we may not have observations of the wear or tear on IoT devices, or their sensing quality. Therefore, to reach the potential of DRL, there are some key challenges to overcome. First, a DRL agent should be able to discern and re-distribute the delayed rewards, such as the backwards-view of tasks in [179]. Second, the simulation models of partially observable environments should be built to be non-stationary so at to encourage DRL agents to learn policies that adapt to this non-stationarity. Some recent studies on meta-learning have focused on this area [180].

D. Huge Cost of Trial-and-Error in Online Learning

After deploying in real-world scenarios, the experience-driven DRL can continuously evolve its policy through accumulating new experiences and trial-and-error learning. Most of the existing studies on applying DRL to IoT scenarios use ϵ–greedy as the exploration algorithm in trial-and-error learning, which is characterized by taking actions at random under certain probability. However, in many IoT scenarios, the cost of trial-and-error is so significant that it is impractical to naively apply DRL online training in real-world scenarios.

The above challenges have driven a new branch of DRL called safe RL [181]. The core idea of safe RL is to approximate the optimal policy while meeting the safety constraints and maintaining acceptable performance during trial-and-error learning with the following approaches: 1) Modifying the reward function and giving a more stringent optimization criteria (e.g., worst-case criterion [182]), 2) Modifying the exploration policy and introducing external knowledge (e.g., teacher-student learning [183]). However, since the IoT network is relatively complex, designing a safe RL algorithm for such an environment requires human expert knowledge that may not be always available. Thus, it is worthwhile to study how to extract knowledge from the environments and integrate expert knowledge with DRL training.

Another potential solution is the distributed RL training, which distributes the training process to multiple independent environments and accumulates experiences (or knowledge) to obtain the optimal policy. In this respect, the cost of trial-and-error can be scattered across multiple environments, thereby becoming relatively lower from the perspective of a single IoT device. However, this strategy also attracts the following concerns: 1) Distributed RL training requires to upload data (or parameters) to the central parameter server, which introduces vulnerabilities of malicious attacks and privacy leakages. 2) Accumulating experiences (or knowledge) from multiple environments introduces additional communication overhead. 3) There may be selfish agents that scarcely explore in state-action spaces and make no contribution to the model improvement. Therefore, it is expected to fully exploit the power of distributed RL training in the future.

E. Improving Data Efficiency

As mentioned above, most of the recent studies are limited to simulation environments, where training data (or experience) can be generated near unboundedly. However, in real-world scenarios such as IoT networks, collecting training data involves interactions with the environments; the interactions may be quite expensive. To address this issue, some attempts have been made to improve data efficiency for data-limited domains, such as the prioritized experience replay [184] that samples with different weights. Another promising advent for future work is model-based DRL, which tends to be more data-efficient compared with model-free DRL [8]. In literature, it can be found that most of the studies on applying DRL to IoT only concentrate on model-free DRL, which obtains the optimal policy without explicitly learning the model of the environment. This is different from model-based DRL which focuses on training an accurate model to characterize the environment. However, model-based DRL also presents its drawbacks. For example, model-based DRL is more computationally expensive and complex, thereby limiting its applications in the IoT network. Furthermore, it is impractical to train an accurate model to characterize the rapidly-varied environments or too complicated environments. Therefore, it is of great significance to explore application scenarios suitable for model-based DRL in the future.

F. Learning From Noisy Environments

Compared with simulation environments, applying DRL to real-world environments is generally more complicated because it involves noises caused by sensor errors, false information, asynchronous states, etc. Meanwhile, the state noise inevitably leads to perturbed rewards. In particular, cumulative noise in a noisy environment may cause DRL agents to update parameters in the wrong direction, thereby hindering the convergence and degrading the performance. A few attempts have been made to address these challenges. For example, a novel DRL algorithm called G-learning was proposed in [185] by regularizing the estimated reward. A noise tolerance mechanism for DRL agents was proposed in [186], which leverages the historical experience to estimate state noises and thus identify noisy states that may result in mistakes. However, the above studies are still in its infancy and is assessed only in a few simple environments. Nonetheless, they shed light on future research on training DRL agents in noisy and complex environments such as IoT networks.

IX. Conclusion

In this article, we deliver the survey with the comprehensive literature review on the deep reinforcement learning (DRL) in a wide variety of IoT applications. The objective of this survey is to provide a comprehensive state-of-the-art literature review of applying DRL to solve 4Cs problems in IoT and identify several urgency issues that shall be addressed. We first review the state-of-the-art DRL algorithms as well as their important extensions and discuss the issues including their mechanisms, advantages and existing challenges. Second, we provide a comprehensive review of a wide diversity of IoT applications including smart grid, ITS, IIoT, MCS, blockchain-empowered IoT; they also adopt DRL algorithms to solve the corresponding technical challenges. We also provide a comparison and a guideline for using different DRL algorithms in the various IoT domains and applications. Finally, we highlight the challenges and outline future research directions in driving the further success of DRL in IoT applications.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[2] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, and X. Du, "From IoT to 5G I-IoT: The next generation IoT-based intelligent algorithms and 5G technologies," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 114–120, Oct. 2018.

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

[4] X. He, K. Wang, and W. Xu, "QoE-driven content-centric caching with deep reinforcement learning in edge-enabled IoT," *IEEE Comput. Intell. Mag.*, vol. 14, no. 4, pp. 12–20, Nov. 2019.

[5] X. Wang, Y. Gu, Y. Cheng, A. Liu, and C. L. P. Chen, "Approximate policy-based accelerated deep reinforcement learning," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 31, no. 6, pp. 1820–1830, Jun. 2020.

[6] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.

[7] Q. Chen *et al.*, "A survey on an emerging area: Deep learning for smart city data," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 5, pp. 392–410, Oct. 2019.

[8] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process Mag.*, vol. 34, no. 66, pp. 26–38, Nov. 2017.

[9] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[10] 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 Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1722–1760, 3rd Quart., 2020.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[12] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 5, pp. 679–684, 1957.

[13] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, p. 1054, Sep. 1998.

[14] C. J. C. H. Watkins and P. Dayan, "*Q*-learning," in *Machine Learning*. Boston, MA, USA: Kluwer Acad. Publ., 1992, pp. 279–292.

[15] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[16] G. A. Rummery and M. Niranjan, "On-line *Q*-learning using connectionist systems," Dept. Eng., Cambridge Univ., Cambridge, U.K., Rep. 166, 1994.

[17] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double *Q*-learning," 2015. [Online]. Available: arXiv:1509.06461.

[18] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3, pp. 229–256, May 1992.

[19] V. Konda, "Actor-critic algorithms," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2002.

[20] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," 2016. [Online]. Available: arXiv:1602.01783.

[21] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. A. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. ICML*, Jun. 2014, pp. 387–395.

[22] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: arXiv:1509.02971.

[23] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, no. 1, pp. 181–211, 1999.

[24] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. B. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," 2016. [Online]. Available: arXiv:1604.06057.

[25] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[26] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," 2018. [Online]. Available: arXiv:1802.05438.

[27] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," 2017. [Online]. Available: arXiv:1706.02275.

[28] S. Kim and H. Lim, "Reinforcement learning based energy management algorithm for smart energy buildings," *Energies*, vol. 11, no. 8, p. 2010, 2018.

[29] 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 Things J.*, vol. 6, no. 5, pp. 7360–7374, Oct. 2019.

[30] Y. Li, X. Zhao, and H. Liang, "Throughput maximization by deep reinforcement learning with energy cooperation for renewable ultra-dense IoT networks," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9091–9102, Sep. 2020.

[31] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2013.

[32] F. Li, J. Qin, and W. X. Zheng, "Distributed *Q*-learning-based online optimization algorithm for unit commitment and dispatch in smart grid," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 4146–4156, Sep. 2020.

[33] L. Xiao, X. Xiao, C. Dai, M. Peng, L. Wang, and H. V. Poor, "Reinforcement learning-based energy trading for microgrids," 2018. [Online]. Available: arXiv:1801.06285.

[34] G. Gao, J. Li, and Y. Wen, "DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8472–8484, Sep. 2020.

[35] E. Foruzan, L.-K. Soh, and S. Asgarpoor, "Reinforcement learning approach for optimal distributed energy management in a microgrid," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5749–5758, Sep. 2018.

[36] W. Liu, P. Zhuang, H. Liang, J. Peng, and Z. Huang, "Distributed economic dispatch in microgrids based on cooperative reinforcement learning," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 29, no. 6, pp. 2192–2203, Jun. 2018.

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

[38] H. Shirzeh, F. Naghdy, P. Ciufo, and M. Ros, "Balancing energy in the smart grid using distributed value function (DVF)," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 808–818, Mar. 2015.

[39] H. Xu, H. Sun, D. Nikovski, S. Kitamura, K. Mori, and H. Hashimoto, "Deep reinforcement learning for joint bidding and pricing of load serving entity," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6366–6375, Nov. 2019.

[40] R. Lu, S. H. Hong, and X. Zhang, "A dynamic pricing demand response algorithm for smart grid: Reinforcement learning approach," *Appl. Energy*, vol. 220, pp. 220–230, Jun. 2018.

[41] R. Lu and S. H. Hong, "Incentive-based demand response for smart grid with reinforcement learning and deep neural network," *Appl. Energy*, vol. 236, pp. 937–949, Feb. 2019.

[42] A. Ghasemkhani and L. Yang, "Reinforcement learning based pricing for demand response," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[43] B.-G. Kim, Y. Zhang, M. van der Schaar, and J.-W. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2187–2198, Sep. 2016.

[44] H. Wang, T. Huang, X. Liao, H. Abu-Rub, and G. Chen, "Reinforcement learning for constrained energy trading games with incomplete information," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3404–3416, Oct. 2017.

[45] P. Huang, A. Scheller-Wolf, and K. Sycara, "Design of a multi-unit double auction E-market," *Comput. Intell.*, vol. 18, no. 4, pp. 596–617, 2002.

[46] L. A. Hurtado, E. Mocanu, P. H. Nguyen, M. Gibescu, and R. I. G. Kamphuis, "Enabling cooperative behavior for building demand response based on extended joint action learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 127–136, Jan. 2018.

[47] X. Wang, M. Zhang, and F. Ren, "A hybrid-learning based broker model for strategic power trading in smart grid markets," *Knowl. Based Syst.*, vol. 119, pp. 142–151, Mar. 2017.

[48] Y. Yang, J. Hao, Z. Wang, M. Sun, and G. Strbac, "Recurrent deep multiagent *Q*-learning for autonomous agents in future smart grid," in *Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, Richland, SC, USA, 2018, pp. 2136–2138.

[49] Y. Su, X. Lu, Y. Zhao, L. Huang, and X. Du, "Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9561–9569, Oct. 2019.

[50] C. Savaglio, P. Pace, G. Aloi, A. Liotta, and G. Fortino, "Lightweight reinforcement learning for energy efficient communications in wireless sensor networks," *IEEE Access*, vol. 7, pp. 29355–29364, 2019.

[51] S. Chinchali *et al.*, "Cellular network traffic scheduling with deep reinforcement learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 766–774.

[52] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless network," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[53] Z. Wei, F. Liu, Z. Lyu, X. Ding, L. Shi, and C. Xia, "Reinforcement learning for a novel mobile charging strategy in wireless rechargeable sensor networks," in *Wireless Algorithms, Systems, and Applications*, S. Chellappan, W. Cheng, and W. Li, Eds. Cham, Switzerland: Springer, 2018, pp. 485–496.

[54] X. He, H. Jiang, Y. Song, C. He, and H. Xiao, "Routing selection with reinforcement learning for energy harvesting multi-hop CRN," *IEEE Access*, vol. 7, pp. 54435–54448, 2019.

[55] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (DDPG) based energy harvesting wireless communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8577–8588, Oct. 2019.

[56] M. Chincoli, S. Stavrou, and A. Liotta, "Density and transmission power in intelligent wireless sensor networks," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 1518–1523.

[57] C. Wu, Y. Wang, and Z. Yin, "Energy-efficiency opportunistic spectrum allocation in cognitive wireless sensor network," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 13, Jan. 2018.

[58] J. Yan, H. He, X. Zhong, and Y. Tang, "*Q*-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 200–210, 2017.

[59] C. B. Browne *et al.*, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[60] Z. Ni and S. Paul, "A multistage game in smart grid security: A reinforcement learning solution," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 30, no. 9, pp. 2684–2695, Sep. 2019.

[61] J. Duan *et al.*, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 814–817, Jan. 2020.

[62] Y. Wadhawan and C. Neuman, "Rl-bags: A tool for smart grid risk assessment," in *Proc. Int. Conf. Smart Grid Clean Energy Technol. (ICSGCE)*, May 2018, pp. 7–14.

[63] M. Panfili, A. Giuseppi, A. Fiaschetti, H. B. Al-Jibreen, A. Pietrabissa, and F. D. Priscoli, "A game-theoretical approach to cyber-security of critical infrastructures based on multi-agent reinforcement learning," in *Proc. 26th Mediterr. Conf. Control Autom. (MED)*, Jun. 2018, pp. 460–465.

[64] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang, "Online cyber-attack detection in smart grid: A reinforcement learning approach," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5174–5185, Sep. 2019.

[65] A. O. Otuoze, M. W. Mustafa, and R. M. Larik, "Smart grids security challenges: Classification by sources of threats," *J. Electr. Syst. Inf. Technol.*, vol. 5, no. 3, pp. 468–483, 2018.

[66] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, "Evaluation of reinforcement learning-based false data injection attack to automatic voltage control," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2158–2169, Mar. 2019.

[67] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4592–4601, Jun. 2017.

[68] M. Yan, G. Feng, J. Zhou, and S. Qin, "Smart multi-RAT access based on multiagent reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4539–4551, May 2018.

[69] C. Wu, C. Wang, J. Sheng, and Y. Wang, "Cooperative learning for spectrum management in railway cognitive radio network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5809–5819, Jun. 2019.

[70] 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 Commun. Mag.*, vol. 57, no. 11, pp. 102–108, Nov. 2019.

[71] 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 Trans. Wireless Commun.*, vol. 19, no. 1, pp. 507–519, Jan. 2020.

[72] F. Li, X. Song, H. Chen, X. Li, and Y. Wang, "Hierarchical routing for vehicular ad hoc networks via reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1852–1865, Feb. 2019.

[73] B. T. Sharef, R. A. Alsaqour, and M. Ismail, "Review: Vehicular communication ad hoc routing protocols: A survey," *J. Netw. Comput. Appl.*, vol. 40, pp. 363–396, Apr. 2014.

[74] H. Yang, X. Xie, and M. Kadoch, "Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4157–4169, May 2019.

[75] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[76] U. Challita, W. Saad, and C. Bettstetter, "Cellular-connected UAVs over 5G: Deep reinforcement learning for interference management," 2018. [Online]. Available: arXiv:1801.05500.

[77] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Apr. 2019.

[78] C. Gallicchio and A. Micheli, "Deep echo state network (DeepESN): A brief survey," 2017. [Online]. Available: arXiv:1712.04323.

[79] L. Xiao, C. Xie, M. Min, and W. Zhuang, "User-centric view of unmanned aerial vehicle transmission against smart attacks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3420–3430, Apr. 2018.

[80] C. Li, W. Zhou, K. Yu, L. Fan, and J. Xia, "Enhanced secure transmission against intelligent attacks," *IEEE Access*, vol. 7, pp. 53596–53602, 2019.

[81] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[82] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H. V. Poor, "Two-dimensional antijamming mobile communication based on reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9499–9512, Oct. 2018.

[83] S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10450–10464, Oct. 2020.

[84] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[85] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[86] J. Yao and N. Ansari, "Caching in dynamic IoT networks by deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3268–3275, Mar. 2021.

[87] G. Dulac-Arnold *et al.*, "Deep reinforcement learning in large discrete action spaces," 2015. [Online]. Available: arXiv:1512.07679.

[88] T. Zahavy, M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor, "Learn what not to learn: Action elimination with deep reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3562–3573.

[89] T. Sen and H. Shen, "Machine learning based timeliness-guaranteed and energy-efficient task assignment in edge computing systems," in *Proc. IEEE 3rd Int. Conf. Fog Edge Comput. (ICFEC)*, May 2019, pp. 1–10.

[90] H. P. Sajjad, K. Danniswara, A. Al-Shishtawy, and V. Vlassov, "SpanEdge: Towards unifying stream processing over central and near-the-edge data centers," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 168–178.

[91] Z. Dong *et al.*, "An energy-efficient offloading framework with predictable temporal correctness," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, 2017, p. 19.

[92] 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 Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

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

[94] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Partially observable reinforcement learning for intelligent transportation systems," 2018. [Online]. Available: arXiv:1807.01628.

[95] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[96] Y. Liu, L. Liu, and W. Chen, "Intelligent traffic light control using distributed multi-agent Q learning," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[97] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[98] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1274–1285, Jun. 2020.

[99] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3204–3217, Nov. 2017.

[100] C. Wang, J. Zhang, L. Xu, L. Li, and B. Ran, "A new solution for freeway congestion: Cooperative speed limit control using distributed reinforcement learning," *IEEE Access*, vol. 7, pp. 41947–41957, 2019.

[101] C. You, J. Lu, D. Filev, and P. Tsiotras, "Autonomous planning and control for intelligent vehicles in traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2339–2349, Jun. 2020.

[102] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2156–2162.

[103] J. J. Q. Yu, W. Yu, and J. Gu, "Online vehicle routing with neural combinatorial optimization and deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3806–3817, Oct. 2019.

[104] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1262–1276, Jun. 2019.

[105] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[106] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud rans," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.

[107] H. Fan, L. Zhu, C. Yao, J. Guo, and X. Lu, "Deep reinforcement learning for energy efficiency optimization in wireless networks" in *Proc. IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2019, pp. 465–471.

[108] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019.

[109] B. Demirel, A. Ramaswamy, D. E. Quevedo, and H. Karl, "DeepCAS: A deep reinforcement learning algorithm for control-aware scheduling," *IEEE Control Syst. Lett.*, vol. 2, no. 4, pp. 737–742, Oct. 2018.

[110] S. S. Oyewobi, G. P. Hancke, A. M. Abu-Mahfouz, and A. J. Onumanyi, "An effective spectrum handoff based on reinforcement learning for target channel selection in the industrial Internet of Things," *Sensors*, vol. 19, no. 6, p. 1395, 2019.

[111] 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 Things J.*, vol. 5, no. 4, pp. 2375–2385, Aug. 2018.

[112] Z. Hong, W. Chen, H. Huang, S. Guo, and Z. Zheng, "Multi-hop cooperative computation offloading for industrial IoT-edge-cloud computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 12, pp. 2759–2774, Dec. 2019.

[113] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, early access, Mar. 4, 2019, doi: 10.1109/TETC.2019.2902661.

[114] H. He, H. Shan, A. Huang, Q. Ye, and W. Zhuang, "Reinforcement learning-based computing and transmission scheduling for LTE-U-enabled IoT," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[115] J. Wang, Y. Cao, J. Liu, and Y. Zhang, "Deep reinforcement learning based task offloading in SDN-enabled industrial Internet of Things," in *Proc. Int. Conf. Artif. Intell. Commun. Netw.*, 2019, pp. 425–437.

[116] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance optimization in mobile-edge computing via deep reinforcement learning," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, 2018, pp. 1–6.

[117] 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 Things J.*, vol. 6, no. 4, pp. 7011–7024, Aug. 2019.

[118] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019.

[119] J. Huang, Q. Chang, and N. Chakraborty, "Machine preventive replacement policy for serial production lines based on reinforcement learning," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 523–528.

[120] S. R. A. Barde, S. Yacout, and H. Shin, "Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks," *J. Intell. Manuf.*, vol. 30, no. 1, pp. 147–161, 2019.

[121] X. Wang, H. Wang, and C. Qi, "Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system," *J. Intell. Manuf.*, vol. 27, no. 2, pp. 325–333, Apr. 2016.

[122] E. Tuncel, A. Zeid, and S. Kamarthi, "Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning," *J. Intell. Manuf.*, vol. 25, no. 4, pp. 647–659, Aug. 2014.

[123] A. Ferdowsi, U. Challita, W. Saad, and N. B. Mandayam, "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 307–312.

[124] P. M. Shakeel, S. Baskar, V. R. S. Dhulipala, S. Mishra, and M. M. Jaber, "Maintaining security and privacy in health care system using learning based deep-Q-networks," *J. Med. Syst.*, vol. 42, p. 186, Aug. 2018.

[125] R. Meyes **et al.**, "Motion planning for industrial robots using reinforcement learning," *Procedia CIRP*, vol. 63, pp. 107–112, 2017.

[126] G. Schoettler *et al.*, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," 2019. [Online]. Available: arXiv:1906.05841.

[127] M. Geng, X. Zhou, B. Ding, H. Wang, and L. Zhang, "Learning to cooperate in decentralized multi-robot exploration of dynamic environments," in *Neural Information Processing*, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds. Cham, Switzerland: Springer, 2018, pp. 40–51.

[128] D. Schwung, F. Csaplar, A. Schwung, and S. X. Ding, "An application of reinforcement learning algorithms to industrial multi-robot stations for cooperative handling operation," in *Proc. IEEE 15th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2017, pp. 194–199.

[129] S. Zheng, C. Gupta, and S. Serita, "Manufacturing dispatching using reinforcement and transfer learning," Oct. 2019. [Online]. Available: arXiv:1910.02035

[130] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop Hot Topics Netw.*, 2016, pp. 50–56.

[131] H. Chen, X. Li, and F. Zhao, "A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2763–2774, Apr. 2016.

[132] M. N. Jambli, M. I. Bandan, K. S. Pillay, and S. M. Suhaili, "An analytical study of leach routing protocol for wireless sensor network," in *Proc. IEEE Conf. Wireless Sens. (ICWiSe)*, Nov. 2018, pp. 44–49.

[133] L. Cai, M. Boukhechba, N. Kaur, C. Wu, L. E. Barnes, and M. S. Gerber, "Adaptive passive mobile sensing using reinforcement learning," in *Proc. IEEE 20th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Jun. 2019, pp. 1–6.

[134] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1535–1545, Feb. 2018.

[135] S. Kim, "Effective crowdsensing and routing algorithms for next generation vehicular networks," *Wireless Netw.*, vol. 25, no. 4, pp. 1815–1827, 2019.

[136] G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntés-Mulero, and A. Cabellos, "A deep-reinforcement learning approach for software-defined networking routing optimization," 2017. [Online]. Available: arXiv:1709.07080.

[137] W. Huang, Y. Wang, and X. Yi, "A deep reinforcement learning approach to preserve connectivity for multi-robot systems," in *Proc. 10th Int. Congr. Image Signal Process. BioMed. Eng. Informat. (CISP-BMEI)*, Oct. 2017, pp. 1–7.

[138] Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang, "A context-aware multi-armed bandit incentive mechanism for mobile crowd sensing systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7648–7658, Oct. 2019.

[139] N. Eagle and D. Lazer, "Inferring social network structure using mobile phone data," *Proc. Nat. Acad. Sci.*, vol. 106, Jan. 2007.

[140] B. Guo, C. Chen, D. Zhang, Z. Yu, and A. Chin, "Mobile crowd sensing and computing: When participatory sensing meets participatory social media," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 131–137, Feb. 2016.

[141] Y. Zhan, Y. Xia, J. Zhang, T. Li, and Y. Wang, "Crowdsensing game with demand uncertainties: A deep reinforcement learning approach," 2019. [Online]. Available: arXiv:1901.00733.

[142] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *J. Finan.*, vol. 16, no. 1, pp. 8–37, 1961.

[143] Y. Chen and H. Wang, "IntelligentCrowd: Mobile crowdsensing via multi-agent reinforcement learning," 2018. [Online]. Available: arXiv:1809.07830.

[144] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2316–2329, Oct. 2020.

[145] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, and W. Wang, "Learning-based energy-efficient data collection by unmanned vehicles in smart cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1666–1676, Apr. 2018.

[146] C. Liu, Z. Dai, Y. Zhao, J. Crowcroft, D. O. Wu, and K. K. Leung, "Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 130–146, Jan. 2021.

[147] L. Wang, W. Liu, D. Zhang, Y. Wang, E. Wang, and Y. Yang, "Cell selection with deep reinforcement learning in sparse mobile crowdsensing," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1543–1546.

[148] L. Wang *et al.* "SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, p. 20, Oct. 2017.

[149] W. Liu, Y. Yang, E. Wang, L. Wang, D. Zeghlache, and D. Zhang, "Multi-dimensional urban sensing in sparse mobile crowdsensing," *IEEE Access*, vol. 7, pp. 82066–82079, 2019.

[150] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, p. 21, Apr. 2019.

[151] F. B. Mismar and B. L. Evans, "Deep *Q*-learning for self-organizing networks fault management and radio performance improvement," 2017. [Online]. Available: arXiv:1707.02329.

[152] A. Ferdowsi and W. Saad, "Deep learning-based dynamic watermarking for secure signal authentication in the Internet of Things," 2017. [Online]. Available: arXiv:1711.013.

[153] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Netw.*, vol. 32, no. 4, pp. 54–60, Jul./Aug. 2018.

[154] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 35–47, 2018.

[155] M. Zhang, J. Chen, L. Yang, and J. Zhang, "Dynamic pricing for privacy-preserving mobile crowdsensing: A reinforcement learning approach," *IEEE Netw.*, vol. 33, no. 2, pp. 160–165, Mar./Apr. 2019.

[156] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[157] E. A. Mazied *et al.*, "The wireless control plane: An overview and directions for future research," *J. Netw. Comput. Appl.*, vol. 126, pp. 104–122, Jan. 2019.

[158] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, p. 970, 2019.

[159] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks: A survey," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 102–110, Feb. 2016.

[160] R. Venkataraman, S. Moeller, B. Krishnamachari, and T. R. Rao, "Trust-based backpressure routing in wireless sensor networks," *Int. J. Sens. Netw.*, vol. 17, no. 1, pp. 27–39, 2015.

[161] J. Gao, Y. Shen, M. Ito, and N. Shiratori, "Multi-agent *Q*-learning aided backpressure routing algorithm for delay reduction," 2017. [Online]. Available: arXiv:1708.06926.

[162] N. C. Luong, T. T. Anh, H. T. T. Binh, D. Niyato, D. I. Kim, and Y.-C. Liang, "Joint transaction transmission and channel selection in cognitive radio based blockchain networks: A deep reinforcement learning approach," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 8409–8413.

[163] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, "Blockchain and deep reinforcement learning empowered intelligent 5G beyond," *IEEE Netw.*, vol. 33, no. 3, pp. 10–17, May/Jun. 2019.

[164] L. Xiao *et al.*, "A reinforcement learning and blockchain-based trust mechanism for edge networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5460–54701, Sep. 2020.

[165] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, Aug. 2019.

[166] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019.

[167] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial Internet of Things: A dueling deep *Q*-learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, Jun. 2019.

[168] M. Liu, Y. Teng, F. R. Yu, V. C. Leung, and M. Song, "Deep reinforcement learning based performance optimization in blockchain-enabled Internet of Vehicle," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[169] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 6, pp. 45–58, Oct. 2007.

[170] P.-L. Aublin, R. Guerraoui, N. Knežević, V. Quéma, and M. Vukolić, "The next 700 BFT protocols," *ACM Trans. Comput. Syst.*, vol. 32, no. 4, p. 12, Jan. 2015.

[171] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3559–3570, Jun. 2019.

[172] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3516–3526, Jun. 2019.

[173] A. Outchakoucht, E. Hamza, and J. P. Leroy, "Dynamic access control policy based on blockchain and machine learning for the Internet of Things," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, pp. 417–424, 2017.

[174] L. Xiao, T. Chen, G. Han, W. Zhuang, and L. Sun, "Game theoretic study on channel-based authentication in MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7474–7484, Aug. 2017.

[175] C. Dai, X. Xiao, Y. Ding, L. Xiao, Y. Tang, and S. Zhou, "Learning based security for VANET with blockchain," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, 2018, pp. 210–215.

[176] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7992–8004, Oct. 2019.

[177] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive Internet-of-Things systems," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1371–1387, Feb. 2019.

[178] C. S. de Witt, J. Foerster, G. Farquhar, P. Torr, W. Boehmer, and S. Whiteson, "Multi-agent common knowledge reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9927–9939.

[179] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter, "RUDDER: Return decomposition for delayed rewards," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13566–13577.

[180] A. Nagabandi, C. Finn, and S. Levine, "Deep online learning via meta-learning: Continual adaptation for model-based RL," 2018. [Online]. Available: arXiv:1812.07671.

[181] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 42, pp. 1437–1480, 2015.

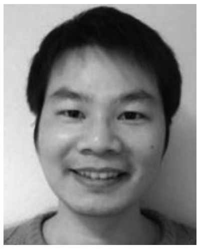
[182] A. Tamar, H. Xu, and S. Mannor, "Scaling up robust MDPs by reinforcement learning," 2013. [Online]. Available: arXiv:1306.6189.

[183] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, 2015, pp. 1–9.

[184] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016.

[185] R. Fox, A. Pakman, and N. Tishby, "Taming the noise in reinforcement learning via soft updates," 2015. [Online]. Available: arXiv:1512.08562.

[186] R. Ribeiro, A. L. Koerich, and F. Enembreck, "Noise tolerance in reinforcement learning algorithms," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. (IAT)*, Nov. 2007, pp. 265–268.



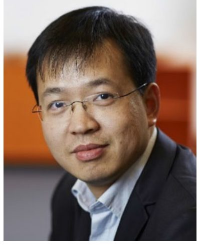
Wuhui Chen (Member, IEEE) received the bachelor's degree from Northeast University, Shenyang, China, in 2008, and the master's and Ph.D. degrees from the University of Aizu, Aizu–Wakamatsu, Japan, in 2011 and 2014, respectively. From 2014 to 2016, he was a Research Fellow with the Japan Society for the Promotion of Science, Japan. From 2016 to 2017, he was a Researcher with the University of Aizu. He is currently an Associate Professor with Sun Yat-Sen University, Guangzhou, China. His research interests include edge/cloud computing, cloud robotics, and blockchain.

Xiaoyu Qiu received the B.S. degree from Sun Yat-Sen University, Guangzhou, China, in 2020, where he is currently pursuing the M.S. degree with the School of Computer Science and Engineering. He is proactively working on edge computing, cloud computing, cloud robotics, and computation offloading, with emphasis on artificial intelligence in edge/cloud computing.

Ting Cai is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. From 2012 to 2018, she was a Lecturer with the College of Mobile Telecom, Chongqing University of Posts and Telecom, China. Her current research interests include blockchain, reinforcement learning, Internet of Things security, and edge/cloud computing.

Hong-Ning Dai (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, Chinese University of Hong Kong. He is currently an Associate Professor with the Faculty of Information Technology, Macau University of Science and Technology. His current research interests include the Internet of Things and blockchain technology. He has served as an Associate Editor for IEEE ACCESS, and a Guest Editor for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING.

Zibin Zheng (Senior Member, IEEE) is a Professor and the Deputy Dean with the School of Software Engineering, Sun Yat-sen University, China. He published over 200 international journal and conference papers, including one ESI Hot Paper and six ESI highly cited papers. According to Google Scholar, his papers have more than 15 000 citations. His research interests include blockchain, software engineering, and services computing. He was a recipient of several awards, including the Top 50 Influential Papers in Blockchain of 2018, the ACM SIGSOFT Distinguished Paper Award at ICSE2010, and the Best Student Paper Award at ICWS2010. He served as the BlockSys'19 and CollaborateCom'16 General Co-Chair, SC2'19, ICIOT'18 and IoV'14 PC Co-Chair. He is a Fellow of the IET.



Yan Zhang (Fellow, IEEE) received the B.S. degree from the Nanjing University of Post and Telecommunications, the M.S. degree from the Beihang University, and the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is currently a Full Professor with the Department of Informatics, University of Oslo, Norway. His research interests include next-generation wireless networks leading to 6G, green, and secure cyber-physical systems (e.g., smart grid and transport). In 2018, he was a recipient of the Global Highly Cited Researcher Award (Web of Science top 1% most cited worldwide). He is the Symposium/Track Chair in a number of conferences, including IEEE ICC 2021, IEEE SmartGridComm 2021, and IEEE Globecom 2017. He is the Chair of IEEE Communications Society Technical Committee on Green Communications and Computing. He is an Editor (or Area Editor, Senior Editor, Associate Editor) for several IEEE transactions/magazine, including *IEEE Network Magazine*, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE COMMUNICATIONS SURVEY AND TUTORIALS, IEEE INTERNET OF THINGS JOURNAL, IEEE SYSTEMS JOURNAL, *IEEE Vehicular Technology Magazine*, and IEEE Blockchain Technical Briefs. He is an IEEE Communications Society Distinguished Lecturer and an IEEE Vehicular Technology Society Distinguished Speaker. He was an IEEE Vehicular Technology Society Distinguished Lecturer from 2016 to 2020. He is a CCF Senior Member, an Elected Member of CCF Technical Committee of Blockchain, and the 2019 CCF Distinguished Speaker. He is a Fellow of IET, an Elected Member of Academia Europaea, Royal Norwegian Society of Sciences and Letters (DKNVS), and Norwegian Academy of Technological Sciences.