

Distributive Dynamic Spectrum Access Through Deep Reinforcement Learning: A Reservoir Computing-Based Approach

Hao-Hsuan Chang, Hao Song[✉], *Student Member, IEEE*, Yang Yi, *Senior Member, IEEE*, Jianzhong Zhang, *Fellow, IEEE*, Haibo He[✉], *Fellow, IEEE*, and Lingjia Liu[✉], *Senior Member, IEEE*

Abstract—Dynamic spectrum access (DSA) is regarded as an effective and efficient technology to share radio spectrum among different networks. As a secondary user (SU), a DSA device will face two critical problems: 1) avoiding causing harmful interference to primary users (PUs) and 2) conducting effective interference coordination with other SUs. These two problems become even more challenging for a distributed DSA network where there is no centralized controllers for SUs. In this paper, we investigate communication strategies of a distributive DSA network under the presence of spectrum sensing errors. To be specific, we apply the powerful machine learning tool, deep reinforcement learning (DRL), for SUs to learn “appropriate” spectrum access strategies in a distributed fashion assuming NO knowledge of the underlying system statistics. Furthermore, a special type of recurrent neural network, called the reservoir computing (RC), is utilized to realize DRL by taking advantage of the underlying temporal correlation of the DSA network. Using the introduced machine learning-based strategy, SUs could make spectrum access decisions distributedly relying only on their own current and past spectrum sensing outcomes. Through extensive experiments, our results suggest that the RC-based spectrum access strategy can help the SU to significantly reduce the chances of collision with PUs and other SUs. We also show that our scheme outperforms the myopic method which assumes the knowledge of system statistics, and converges faster than the *Q*-learning method when the number of channels is large.

Index Terms—Deep *Q*-network (DQN), deep reinforcement learning (DRL), dynamic spectrum access (DSA), echo state network (ESN), reservoir computing (RC), resource allocation.

I. INTRODUCTION

ACCORDING to CISCO [1], global mobile data traffic has experienced boosting growth, which will increase seven-fold between 2016 and 2021 with a compound annual growth

Manuscript received May 1, 2018; revised July 27, 2018; accepted September 14, 2018. Date of publication September 27, 2018; date of current version May 8, 2019. This work was supported by the National Science Foundation under Grant ECCS-1731672 and Grant ECCS-1811497. (Corresponding author: Lingjia Liu.)

H.-H. Chang, H. Song, Y. Yi, and L. Liu are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: ljiu@vt.edu).

J. Zhang is with the Standards and Mobility Innovation Lab., Samsung Research America, Richardson, TX 75081 USA.

H. He is with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, South Kingstown, RI 02881 USA. Digital Object Identifier 10.1109/JIOT.2018.2872441

rate of 46%. Spectrum extension is critical for future wireless communication networks to cope with this exponential data traffic growth. However, radio spectrum is a costly and scarce resource and the use of radio spectrum is usually regulated by governmental entities. The current shortage of radio spectrum makes it hard for wireless operators to obtain sufficient licensed bands with exclusive ownership. On the other hand, experimental tests and investigations from both academia and industries show that the static spectrum allocation policy from federal communications commission (FCC) causes the under-utilization of allocated licensed bands [2]–[4]. To be specific, according to the report from Shared Spectrum Company, the utilization of most licensed spectrum bands is under 30%, and more than half of licensed spectrum bands are under 20% utilized. For example, even in the most crowded cities, like Washington, DC and Chicago, only 38% of licensed spectrum bands are occupied [5]. These investigations and statistics reflect the fact that radio spectrum resources are under-utilized, which motivates FCC to reconsider the current static spectrum allocation policy, and employ dynamic spectrum access (DSA) to promote spectrum utilizations [6].

A large quantity of public unlicensed frequency bands, such as the industrial, scientific, and medical (ISM) band and the unlicensed national information infrastructure band, were freed up, on which any wireless devices are allowed to seek for spectrum extension using opportunistic spectrum access. However, nowadays, these bands are becoming extremely crowded and over-utilized. To address this issue, ultra-wideband millimeter wave (mmWave) bands have been exploited for unlicensed use, such as 24–24.25 GHz ISM bands and 28–30 GHz local multiple distribution service bands [7]. Moreover, according to the recent spectrum allocation policy released by FCC, 14 GHz consecutive mmWave bands (57–71 GHz) can be used as license-exempt bands [8]. Even though mmWave bands provide a lot of opportunities for radio spectrum access, it is important to note that the use of mmWave band usually requires sophisticated signal processing technologies and hardware.

To provide access opportunities on low frequency bands, FCC opened up licensed spectrum bands to encourage DSA of under-utilized spectrum [6]. For example, in 2015, the FCC conducted an auction of 1695–1710 MHz, 1755–1780 MHz,

and 2155–2180 MHz, also referred to as the Advanced Wireless Services-3 (AWS-3) band, on which DSA devices share the bands with federal systems, such as the federal meteorological-satellite systems [9]. In addition to the AWS-3 band, the 3.5 GHz band has been opened up and utilized by the spectrum access system, in which different types of users will share the spectrum with different spectrum access priority, including incumbent access, priority access, and general authorized access [10]. Undoubtedly, opening up current licensed frequency bands could significantly improve the utilization of radio spectrum. However, many challenges still exist. For example, on opened licensed frequency bands, DSA users should always act as secondary users (SUs) with low priority. This means DSA users should provide protection to primary users (PUs) from harmful interference.

To achieve spectrum co-existence between DSA users and PUs, many strategies have been introduced so far, which can be classified into two main spectrum access mechanisms. The first one is listen-before-talk (LBT), also known as the interweaving scheme, in which an SU can access a frequency band only if it is detected to be available [11]. Although this scheme can effectively avoid strong interference to PUs, opportunities for DSA users to access shared frequency bands can be rather limited. This is because under LBT the spectrum access depends completely on current spectrum sensing outcome. In reality, due to the random nature of wireless environments, limited/no cooperation among SUs, and other practical factors spectrum sensing can never be perfect [12]. This will cause false alarm or miss detection of PUs' activities leading SUs to make inappropriate decisions regarding channel access [12]. The second spectrum access scheme is spectrum sharing, also known as the underlaying scheme [13]. In this scheme, SUs coexist with the PUs on shared frequency bands, and adjust their transmit power level so that the accumulated interference experienced at PUs is less than a tolerable interference threshold. This scheme requires a strong assumption that the channel state information between transmitters of SUs and the receivers of PUs are known as *a priori* in order to conduct power control. However, in reality, it is usually difficult to obtain these channel state information without a central controller. Even under the presence of a central controller, exchanging these channel state information may impose a heavy control overhead for the underlying network making it difficult to be implemented in practice.

In light of these challenges, machine learning-based approach has recently been introduced in the field of DSA due to its ability to adapt to dynamic unknown environments [14]–[16]. To be specific, with machine learning, spectrum access will be determined not only by current spectrum sensing outcome but also by the learning result from past spectrum status. In this way, the negative impact of imperfect spectrum sensing could be greatly alleviated. Additionally, machine learning can enable DSA devices to obtain accurate channel status and useful prediction/statistic information of channel status, such as the behavior of PUs and the load of other SUs, the spectrum access based on which could significantly reduce the collisions between SUs and PUs.

In this paper, we investigate machine learning approaches to obtain artificial intelligence-enabled spectrum access strategies for DSA networks. To reduce the control overhead of the underlying DSA network, we incorporate the powerful machine learning technique, deep reinforcement learning (DRL), for SUs to learn “appropriate” spectrum access strategies in a distributed fashion assuming NO knowledge of the underlying system statistics. Furthermore, a special type of recurrent neural network (RNN), called the reservoir computing (RC), is utilized to realize DRL by taking advantage of the underlying temporal correlation of the DSA network. To be specific, DRL and RC-based DSA scheme is developed to facilitate DSA systems to perform appropriate channel access, aiming at protecting PUs from harmful interference and avoid collisions with other SUs. To the best of our knowledge, this is the first work to combine DRL and RC for DSA networks. DRL aims to solve the large state space problem in traditional reinforcement learning [17]. Traditional reinforcement learning, such as *Q*-learning converges slowly when the state space is large. Meanwhile, DRL utilizes deep neural network to approximate the expected cumulative reward of the state-action pairs and accelerate the convergence time. RC is a special type of RNN that reduces the complexity of training significantly [18]. RNN is a powerful neural network that reasons the *temporal correlation* of input sequences, but the difficulty of training RNN is a well-known trouble to many researchers [17], [19]. RC simplifies the training of RNN by only training the output weights and our previous work in this field including designing a low-complexity and high-efficiency RC-based symbol detector for MIMO-OFDM systems [20]. In this paper, deep *Q*-network (DQN) and RC are combined to design DSA strategies for handling the large state space and taking advantage of the temporal correlation of the underlying wireless channel as well as the PU's activities.

It is important to note that DRL has been applied in [15] and [16] for solving DSA problems. In [15], only one SU is assumed in the network with perfect spectrum sensing outcomes (that is, there is no error in spectrum sensing). On the other hand, [16] assumes no PU in the network, therefore, the collision with PUs is not considered. Furthermore, [15] combines DRL with multilayer perceptron (MLP), which is incapable of learning the temporal correlation of spectrum sensing outcomes. On the other hand, [16] combines DRL with RNN resulting in slow convergence rate, especially in large state space.

In this paper, to better reflect the reality, we assume that there are multiple PUs and multiple SUs in the DSA network by jointly considering effects of colliding with PUs or other SUs. Furthermore, no centralized controller is assumed in our DSA network to allow fully distributed DSA strategies. Imperfect spectrum sensing with unknown sensing error probabilities is also considered in this paper. The main contributions of this paper can be summarized in the following.

- 1) A distributed DSA strategy for SUs of a DSA network is introduced based on the integration of DRL and RC. The strategy takes SUs' imperfect spectrum sensing outcomes into account and enables SUs to conduct spectrum access in a fully distributed fashion.

- 2) Extensive performance evaluation of the introduced DSA strategy has been conducted to show that our machine learning-based spectrum access strategy could quickly learn PUs' activities and significantly reduce the chance of collisions with PUs and SUs in a realistic environment.
- 3) Compared with the myopic scheme which assumes system statistic, our scheme can outperform the myopic strategy in both the single-SU case and the multiple-SU case. Compared with Q -learning, our scheme demonstrates quick convergence rate with better performance. Compared with DRL+MLP, our DRL+RC strategy can take advantage of the underlying temporal correlation of the sensing result to yield substantial performance improvement.

II. DEEP REINFORCEMENT LEARNING AND RESERVOIR COMPUTING

When system dynamics are unknown and observations of the system are not entirely accurate, learning-based methods can adapt to the unknown environment from the partial observation of the system by improving performance on a specific task. Usually, an SU does not know the behaviors of PUs and its spectrum sensor may make a mistake in determining the spectrum holes. Therefore, machine learning is suitable for designing DSA strategies when the system statistics is unknown. In this paper, we adopt DRL and RC, in particular, DQN [21] and echo state network (ESN) [22].

DRL has attracted much attention in recent years because it enables reinforcement learning to efficiently learn in a very large state and action spaces by providing a good approximation of Q -value. Due to the difficulties of training RNN, RC is a new paradigm of RNN training that is simple but powerful. We use a special type of RC, ESN, as the Q -network of DQN.

A. Reinforcement Learning

Reinforcement learning is an important type of machine learning where an agent learns how to behave in an environment to maximize the cumulative reward. It lies between supervised and unsupervised learning. The agent is not told which action to take, which is different from the ground truth labels in supervised learning. Instead, the agent must explore the environment to accumulate knowledge of the environment and exploit its accumulated knowledge to take the best action it can. The agent gradually develops a policy that maps from the observed state to the action. In the distributive DSA work of our interests, each SU is an agent, sensing decision is the action, sensing outcome is the observed state, and sensing strategy is the developed policy.

The most wide-used training technique for reinforcement learning is Q -learning [23]. This technique is a model-free approach that learns the policy directly through interactions with the environment without estimating a model of the environment. Q -learning is a value iteration approach to find the Q -value $Q^\pi(s, a)$ of each state s and action a pairs, which represents the expected value of the cumulative reward when taking action a in the initial state s and then following policy

π thereafter. $Q^\pi(s, a)$ is the Q -value with the optimal policy, and the optimal policy can be derived as

$$\pi^*(s) = \operatorname{argmax}_a Q^\pi(s, a). \quad (1)$$

Assume that the initial state is s_t , an agent takes the action a_t following current policy π and gains a reward r_{t+1} , then enters to the next state s_{t+1} . The objective is to maximize the cumulative discounted reward R

$$R = \sum_{t=1}^{\infty} \gamma^{t-1} r_{t+1}^I \quad (2)$$

where $\gamma \in [0, 1]$ is the discounted rate of the reward. The online update rule of Q -value is

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (3)$$

where $\alpha \in (0, 1)$ is the learning rate and $\gamma \in [0, 1]$ is the discounted rate of the reward. Once the Q -value is updated, the policy π is also updated using ϵ -greedy method [24]

$$a_{t+1} = \begin{cases} \operatorname{argmax}_a Q(s, a), & \text{with probability } 1 - \epsilon \\ \text{Choose random action,} & \text{with probability } \epsilon \end{cases} \quad (4)$$

where ϵ is a small number between 0 and 1. The Q -value and the policy updates iteratively to gradually converge to the optimal policy.

B. Deep Reinforcement Learning

Q -learning performs well for small-scale models but performs poorly for large-scale models. The reason is that the training algorithm of Q -learning iteratively updates the Q -table. As the number of possible states increases, the large Q -table size makes training difficult or even impossible [17].

Due to the difficulties of updating every element in Q -table for a large-scale model, DRL exploits the powerful deep neural network to approximate the Q -value. In 2013, DeepMind team developed a DQN to challenge seven Atari 2600 games and won a human expert on three of the games [25]. In this paper, the size of the state space grows exponentially with the number of the channels. Each channel is occupied by a PU and each PU has two possible states: 1) active state or 2) inactive state, so the state size is 2^N for N channels. This motivates us to use DQN to learn the DSA strategy in an unknown dynamic system.

C. Reservoir Computing

RNN is a powerful network that is capable of learning the temporal behavior for a time sequence. Compared with the MLP, RNN has feedback connections that allow signals to travel in both directions, which exhibits a dynamic behavior. Also, RNN has memory that stores the previous neuron activations. RNN are widely used in the natural language processing because understanding a sentence highly requires temporal relationship between words [19], [26]. Although RNN is more

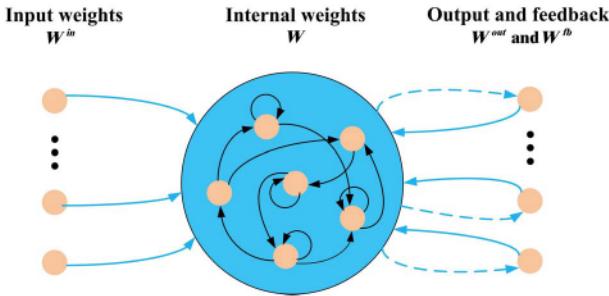


Fig. 1. Reservoir computing.

powerful than the feedforward neural network in terms of the expressiveness, RNN is more challenging to train [17], [27]. The training algorithms for RNN have the vanishing and the exploding gradient problem because the gradient values calculated in long temporal range may become too large or too small, which makes the time for training RNN very long. The aforementioned large state size of channels in DSA further makes training RNN more difficult since we have to model a large-scale dynamic system. Therefore, we need to find an alternative way of training RNN that have faster convergence.

Due to the difficulties of training RNN, RC is introduced to simplify the training with comparable performance [18]. The reservoir is a randomly generated RNN, and it has linear connections to input units and output units, which is shown in Fig. 1. During the training, only the output weights are trained, the weights inside the reservoir and the inputs weights are fixed [28]. The fixed recurrent connections of the reservoir provide a high dimensional dynamics that is able to create all possible spatial and temporal combinations of the input history, which are analogous to cortical dynamics in human brain [29]. The learned output weights determine the best linear combination of the reservoir's state and the input signal to perform the desired task. This approach largely reduces the computation time because only the read out weights are trained. ESN is the most popular type of RC system [20], [30]–[32]. In this paper, we adopt ESN as the Q-network in the DQN.

III. SYSTEM MODEL

We consider a DSA scenario in which primary network consisting of N PUs coexist with DSA secondary network with L SUs. It is assumed that there are totally N wireless channels so that each PU transmit on one unique wireless channel to avoid interference among PUs. Furthermore, we assume that each PU will broadcast warning signals to SUs if the corresponding PU's signal has been collided. To be specific, the warning signal contains information related to which channel has been collided so that the corresponding transmitting SUs on that channel are aware of the collision. This is the only control information from the PU to the SU to enable the protection for the PU. Our proposed method outperforms myopic method that is required to know the system statistics, and converges faster than Q -learning method when the number of channels is large. All wireless channels are shared by all SUs, and an SU will select proper wireless channels to access according to its spectrum access strategy. Moreover, on a wireless channel,

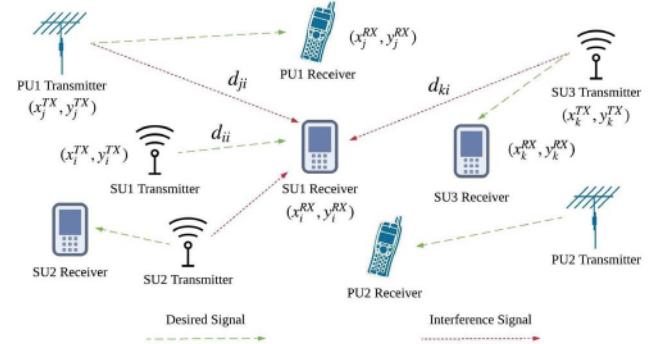


Fig. 2. System model.

an SU should protect the PU, which is authorized to access the channel, from harmful interference.

As shown in Fig. 2, on the X - Y 2-D coordinates, (x_i^{TX}, y_i^{TX}) , (x_i^{RX}, y_i^{RX}) , (x_j^{TX}, y_j^{TX}) , and (x_j^{RX}, y_j^{RX}) , respectively, represent the position coordinates of the transmitter of SU i , the receiver of SU i , the transmitter of PU j , and the receiver of PU j , where $i \in \{1, 2, \dots, L\}$ and $j \in \{1, 2, \dots, N\}$. Accordingly, the distance of a desired signal link can be calculated by $d_{ii} = \sqrt{(x_i^{TX} - x_i^{RX})^2 + (y_i^{TX} - y_i^{RX})^2}$. Moreover, the propagation distance of interference signal from and other SUs are given by $d_{ji} = \sqrt{(x_j^{TX} - x_i^{RX})^2 + (y_j^{TX} - y_i^{RX})^2}$ and $d_{ki} = \sqrt{(x_k^{TX} - x_i^{RX})^2 + (y_k^{TX} - y_i^{RX})^2}$, respectively, where $k \in \{1, 2, \dots, L\}$ and $k \neq i$. The interference only happens when PU j and SU k are using the same wireless channel that SU i is using.

With the propagation distance of d , we adopt the WINNER II channel model [33] to calculate the path loss of the desired signal using

$$PL(d, f_c) = \overline{PL} + A_w \cdot \log_{10}(d[m]) + B_w \cdot \log_{10}\left(\frac{f_c[\text{GHz}]}{5}\right) \quad (5)$$

where f_c is the carrier frequency of wireless channels. \overline{PL} , A_w , and B_w denote the path loss at a reference distance, path loss exponent, and path loss frequency dependence, respectively. Accordingly, the path loss of desired signal $PL(d_{ii}, f_c)$ and interference signals $PL(d_{ji}, f_c)$ and $PL(d_{ki}, f_c)$ can be obtained. It is assumed that a strong line-of-sight (LoS) path exists between a transmitter and a receiver, therefore the Rician channel model is employed to derive channel model, which can be expressed as

$$h = \sqrt{\frac{\kappa}{\kappa + 1}} \sigma e^{j\theta} + \sqrt{\frac{1}{\kappa + 1}} \text{CN}(0, \sigma^2) \quad (6)$$

where $\sigma^2 = 10^{-[(\overline{PL} + A_w \cdot \log_{10}(d[m]) + B_w \cdot \log_{10}((f_c[\text{GHz}]/5)))/(10)]}$ is determined by path loss, and κ is K -factor which indicate the ratio between receiver signal power of an LoS path and scattered paths. $\theta \sim U(0, 1)$ is the phase of arrival signals on LoS path, which takes value from the uniform distribution between 0 and 1, and $\text{CN}(\cdot)$ represents a circularly symmetric complex Gaussian random variable.

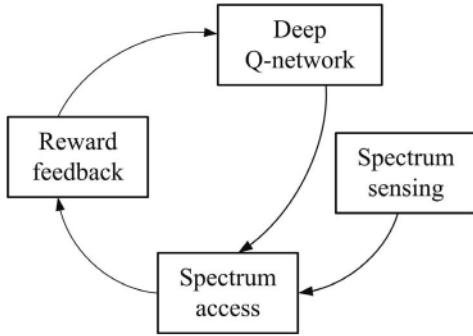


Fig. 3. Learning procedure in DSA.

Thus, the signal to interference plus noise ratio (SINR) of received signals at the receiver of SU i can be derived by

$$\text{SINR}_i = \frac{p_{ij} \cdot |h_{ii}|^2}{p_{jj} \cdot |h_{ji}|^2 + \sum_{k=1, k \neq i}^L p_{kj} \cdot |h_{ki}|^2 + B \cdot N_0} \quad (7)$$

where p_{jj} , p_{ij} , and p_{kj} represent the transmit power of PU j , SU i , and SU k on the j th wireless channel, respectively. $|h_{ii}|^2$, $|h_{ji}|^2$, and $|h_{ki}|^2$, respectively, denote the channel gain of the links between the transmitter i and the receiver i , between the transmitter j and receiver i , and between the transmitter k and receiver i . B and N_0 stand for channel bandwidth and noise spectral density, respectively. With the received SINR of SU i , the channel capacity could be obtained by $C_i = B \cdot \log_2(1 + \text{SINR}_i / \Gamma)$. Here, Γ is the SINR gap that defines the gap between the channel capacity and a practical modulation and coding scheme.

IV. DQN-BASED SPECTRUM ACCESS STRATEGY

In this section, we formulate the distributive DSA as a reinforcement learning problem. We define the agent, state, action, reward, and policy in DSA environment. The learning procedure is shown in Fig. 3. It can be seen that spectrum access strategies are determined by the results of DQN and current spectrum sensing. According to the spectrum access strategies, SUs access wireless channels to carry out data transmissions. Then, SU receives feedback reward based on actual wireless transmission quality, which will be stored by SU transmitters and used as training data of DQN+RC to update spectrum access strategies. The aforementioned learning procedure will be carried out periodically to tackle the variations of wireless environments.

There are a set of $\{1, 2, \dots, N\}$ orthogonal channels and a set of $\{1, 2, \dots, L\}$ SUs. Each channel is occupied by a PU and each PU may be in one of the two states: *Inactive* (1) or *Active* (0). The PU with *Inactive* state means SU is allowed to access the corresponding channel, and the PU with *Active* state means SU cannot access the corresponding channel because the PU is accessing it. The dynamic of each PU's activity is described as a two-state Markov chain as shown in Fig. 4. The transition probability of the two-state Markov chain on the n th

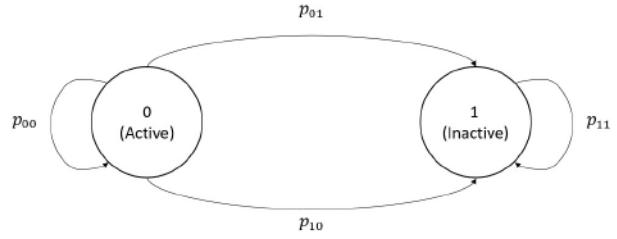


Fig. 4. Two-state Markov chain.

channel is denoted as

$$P_n = \begin{bmatrix} p_{00}^n & p_{01}^n \\ p_{10}^n & p_{11}^n \end{bmatrix} \quad (8)$$

where $p_{ij} = \Pr\{\text{next state is } j | \text{current state is } i\}$ ($i, j \in \{0, 1\}$).

At the beginning of each time slot, every SU performs spectrum sensing on all N channels to detect the channel states. Let the sensing results at time slot t be

$$S(t) = [S^1(t), \dots, S^L(t)] \quad (9)$$

where $S^l(t)$ is an N -dimensional vector $[s_1^l(t), \dots, s_N^l(t)]^T$, and $s_n^l(t) \in \{0, 1\}$ is the sensed state of the l th SU on the n th channel. The spectrum sensing detector is not perfect, so $s_n^l(t)$ may contain error. Let the sensing error probability of the l th SU on the n th channel be E_n^l

$$\Pr\{T_n(t) = s_n^l(t)\} = 1 - E_n^l \quad (10)$$

where $T_n(t)$ is the true state of the n th channel. The transition probabilities and the sensing error probabilities of the channels are both unknown to SUs. The only known information for the l th SU is $S^l(t)$ representing the observed state in the environment and the input of the DQN.

After performing spectrum sensing, each SU decides to access at most one channel or stay idle based on the sensing result. The action of the l th SU is denoted as

$$a^l(t) \in \{0, \dots, N\} \quad (11)$$

where $a^l(t) = n (n > 0)$ means the l th SU decides to access the n th channel at time slot t , and $a^l(t) = 0$ means the l th SU decides not to access any channel at time slot t .

After an SU transmitter accesses a channel, channel states are changed according to its Markov chains. Then the corresponding SU receiver feedbacks the received SINR to the transmitter. As discussed in Section III, the interference will be based on the locations of PUs and SUs. We consider four cases for setting up the reward function.

- 1) An SU accesses a channel that no PU or other SUs are currently using: in this case, the SU does not experience any interference. The achievable data transmission rate, $\log_2(1 + \text{SINR}/\Gamma)$, is used as the reward function.
- 2) An SU accesses a channel that is currently occupied by a PU: in this case, a collision with PU occurs. According to the system model described in Section III the PU will broadcast warning signals to all SUs. Therefore, the corresponding transmitting SUs will be aware of this collision and we set a negative reward of $-C$ ($C > 0$) as an outcome of receiving the warning signal.

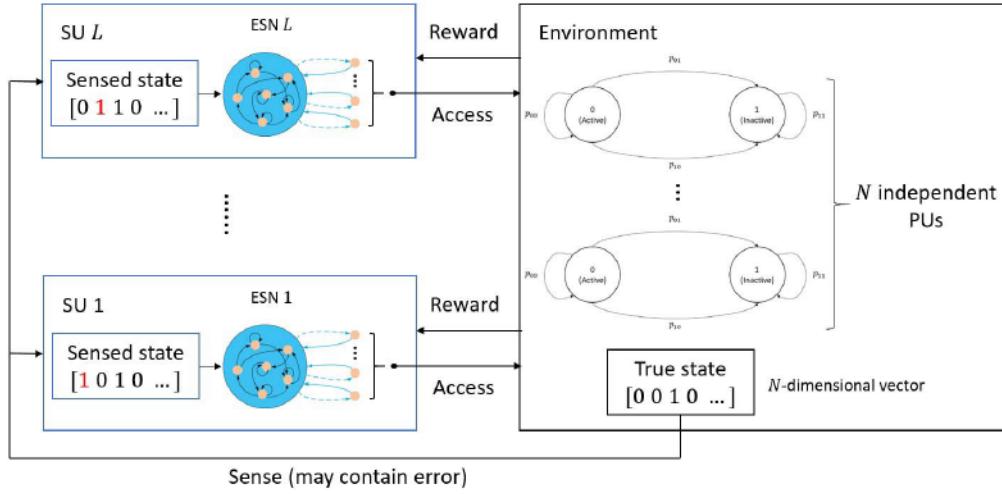


Fig. 5. Architecture for distributive DRL.

- 3) More than two SUs access the available channel: in this case, collisions happen among SUs. Similarly, the achievable data transmission rate, $\log_2(1 + \text{SINR}/\Gamma)$, is used as the reward function.
- 4) An SU decides not to access any channel, and we set the reward to be zero since $\log_2(1 + \text{SINR}/\Gamma) = 0$.

As an outcome, the reward function of the l th SU on the n th channel can be expressed as

$$r^l(t+1) = \begin{cases} -C, & \text{Interference with PU} \\ \log_2(1 + \text{SINR}_n^l/\Gamma), & \text{otherwise.} \end{cases} \quad (12)$$

The DSA strategy is distributive so that the information of sensing results and accessing decisions are not shared between SUs. Each SU has its DQN to make decisions of channel access independently, and the only input to each SU's DQN is sensing results implemented by its sensor. SUs do not know the transition probabilities of channel states and the probabilities of sensing errors. They can only learn how to access the channel through the received SINR that they get after accessing, based on which their access strategies are developed to maximize their own cumulative discounted reward R^l

$$R^l = \sum_{t=1}^{\infty} \gamma^{t-1} r^l(t+1) \quad (13)$$

where $\gamma \in [0, 1]$ denotes the discounted rate. $r^l(t+1)$ is defined in (12). The training method for all DQNs is shown in Algorithm 1.

The architecture of the proposed distributive DSA using DRL and RC is shown in Fig. 5.

V. PERFORMANCE EVALUATION

In this section, we present the experiment results for the distributive DQN-based DSA. We first randomly choose the locations of SUs and PUs in a $150 \text{ m} \times 150 \text{ m}$ space. The distance between a SU transmitter and the corresponding receiver is randomly chosen from 20 m-40 m. As discussed in Section III, the WINNER II model in (5) and the Rician model

Algorithm 1 DQN Training Process

- 1) Initialize DQN_t^l and DQN_e^l with the same structure and initial weights for each SU l .
- 2) Each SU l inputs $S^l(t)$ to DQN_e^l and chooses the action $a^l(t)$ based on Equation (4).
- 3) Each SU l gets a reward r_{t+1}^l , and observes the next state $S^l(t+1)$.
- 4) Store $S^l(t)$, $a^l(t)$, $r^l(t+1)$, $S^l(t+1)$ and replace $S^l(t)$ with $S^l(t+1)$.
- 5) Repeats step 2 to step 4 for T times at T slots.
- 6) Given the stored T sequences, $S^l(t)$, $a^l(t)$, $r^l(t+1)$, $S^l(t+1)$, input $S^l(t)$ and $S^l(t+1)$ to DQN_e^l and DQN_t^l to generate $Q_e^l(S^l(t), a)$ and $Q_t^l(S^l(t+1), a)$, respectively. Then update the $\text{DQN}_e^l(a)$ to minimize the mean-square-error of the following function:

$$[r^l(t+1) + \gamma \max_a Q_t^l(S^l(t+1), a) - Q_e^l(S^l(t), a^l(t))]^2$$

- 8) Replace DQN_t^l with DQN_e^l :

in (6) are employed to calculate the path loss and our derived channel model, respectively. For the WINNER II model, we set $f_c = 5 \text{ GHz}$, $\bar{P}_L = 41$, $A_W = 22.7$, and $B_W = 20$. For the Rician model, we set κ to 8, and σ^2 is determined by the path loss obtained from the WINNER II model. Then the received SINR at an SU's receiver is given by (7), where the bandwidth B is 1 MHz, the noise spectral density N_0 is $10^{-14.7} \text{ (mW/Hz)}$, the transmit power of one SU and is 20 mW, and the transmit power of one PU is 40 mW. All system parameters used in the channel model are listed in Table I.

As described in Section IV, the dynamics of PUs are modelled as independent two-state Markov chains with states: *Inactive* (1) or *Active* (0). To initialize each Markov chain as shown in (8), we randomly choose p_{11} and p_{00} from a uniform distribution over $[0.7, 1]$ and $[0, 0.3]$, respectively, for every channel. Then $p_{10} = 1 - p_{11}$ and $p_{01} = 1 - p_{00}$ can be calculated accordingly. The reason for choosing these ranges for p_{11} and p_{00} is that the utilization of most licensed spectrum

TABLE I
CHANNEL MODEL PARAMETERS

SU transmit power	20mW
PU transmit power	40mW
f_c	5GHz
PL	41
A_W	22.7
B_W	20
κ	8
B	1MHz
N_0	$10^{-14.7}(\text{mW/Hz})$

bands is low, so the possible value of p_{00} should be low, and the possible value of p_{11} should be high.

To demonstrate the effectiveness of our proposed learning method, we employ the myopic and the Q -learning methods as referred methods, which will be compared with learning-based method. In the myopic method, the channel with the maximum immediate reward will be selected [34], [35]. To calculate the expected immediate reward, the myopic method needs the information of the transition probabilities of channels and the probabilities of sensing error. Therefore, we regard the myopic method as a baseline which has more known information than other learning methods, such as DQN and Q -learning. In the myopic method, l th SU first calculates channel state probabilities with $G_n(t) = \Pr(\text{the } n\text{th channel state at time slot } t = 1)$

$$G_n(t) = s_n^l(t) \cdot (1 - E_n^l) + (1 - s_n^l(t)) \cdot E_n^l \quad (14)$$

where $s_n^l(t) \in [0, 1]$ is the sensing result of l th SU on n th channel. E_n^l represents the sensing error probability of l th SU on n th channel. Then the myopic method calculates the expected immediate reward $R_n(t)$ of n th channel

$$R_n(t) = G_n(t) \cdot \left[p_{10}^n \cdot (-C) + p_{11}^n \cdot \log_2 \left(1 + \text{SINR}_n^l / \Gamma \right) \right] + (1 - G_n(t)) \cdot \left[p_{00}^n \cdot (-C) + p_{01}^n \cdot \log_2 \left(1 + \text{SINR}_n^l / \Gamma \right) \right]. \quad (15)$$

Thus, with the myopic method, an SU will access the channel with maximal $R_n(t)$.

On the other hand, Q -learning algorithms have widely been adopted to solve multi-user DSA problems in [36] and [37]. The main drawback of Q -learning is that it is not able to handle a large state size. Since DQN approximates the mapping function from the state space to the Q -value instead of frequently updating Q -table, it has faster convergence than Q -learning when the state space is large.

A. Single SU

In this section, we design experiments to measure the convergence speed of our proposed learning method. In the experiment, we treat the myopic and Q -learning method as the reference. We set the number of SU and channels to 1 and 22, respectively. Note that, the number of PUs is the same as that of channels. Furthermore, the negative reward for colliding with a PU is set to be -2 and the number of neurons of the RC is set to be 64. To ensure a fair comparison, we train both the DQN+RC and Q -learning with the same learning rate of

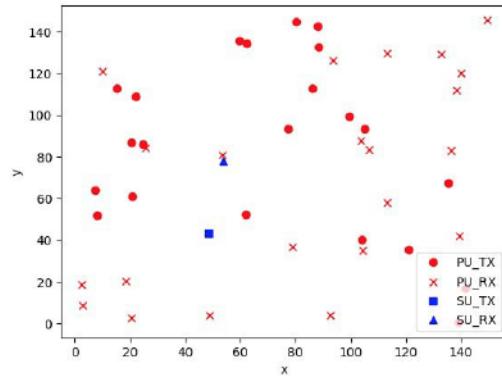


Fig. 6. DSA network geometry (number of channel = 22 and number of SU = 1).

0.01. The locations of PUs and SUs in the underlying DSA network are shown in Fig. 6.

In Fig. 7, we show the average success rate (no collision with PU or SU), the average collision rate with PUs, and the average reward versus the training time. In each iteration, we collect 4000 training sequences ($T = 4000$ in step 5 of Algorithm 1) and calculate the arithmetic mean of the success rate, the collision rate, and the reward. After that we use the trained DQN to determine the access policy for the following iteration.

Although Q -learning and DQN+RC have similar performance when they converge, it can be seen that Q -learning converges much slower than DQN+RC when the number of channels is large. This is because Q -learning needs intensive Q -table updates, and DQN+RC approximates the Q -value using RC, so DQN+RC converges faster if the approximation is accurate. In addition, the results show that DQN+RC and Q -learning can outperform the myopic method even though the myopic method treats transition probabilities of channels and probabilities of sensing error as known information. Therefore, the evaluation result directly suggests that learning-based methods can better adapt to the network dynamics than the myopic method, which assumes more *a priori* information. To sum up, our proposed DQN+RC has the advantages of faster convergence speed when the number of channels is large and better performance than the myopic method.

B. Multiple SUs

Fig. 9 shows the average success rate, the average collision rate with PUs and SUs, and the average reward versus the training time for a DSA network having multiple SUs. The network geometry of the underlying DSA network is shown clearly in Fig. 8. Since SUs may collide with each other, each SU also needs to learn the access strategies of other SUs. We set the number of channels and SUs to be 6 and 2, respectively. The negative reward for colliding with PU is set to -2 as well. For training DQN, we collect 2000 training sequences in each iteration. In this experiment, we also compare with the DQN using MLP as its Q -network. For the MLP scheme, we consider two MLP structures. To be specific, DQN+MLP1

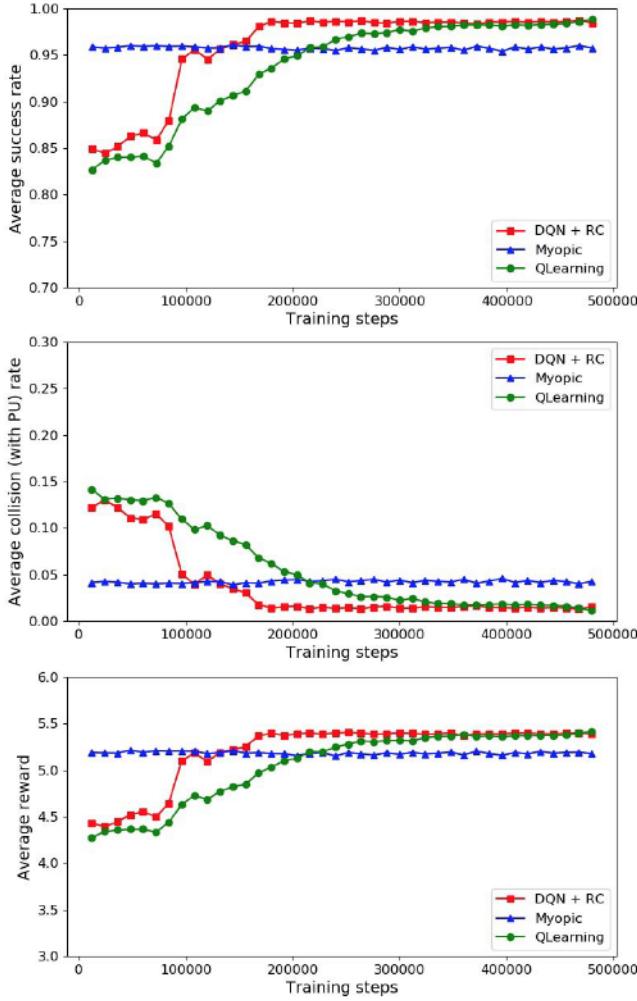


Fig. 7. Average success number, collision number with PUs, and reward with the training time (number of channel = 22 and number of SU = 1).

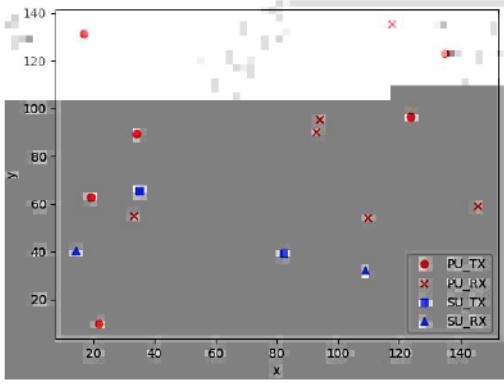


Fig. 8. DSA network geometry (number of channel = 6 and number of SU = 2).

has an MLP with one hidden layer, and DQN+MLP2 has an MLP with two hidden layers. All the hidden layers contain 64 neurons, and the number of neurons of RC is also set to 64. The learning rates of DQN+RC, DQN+MLP1, DQN+MLP2, and *Q*-learning are set to 0.01.

We can observe from Fig. 9 that the myopic method has better performance than all learning-based approaches at the

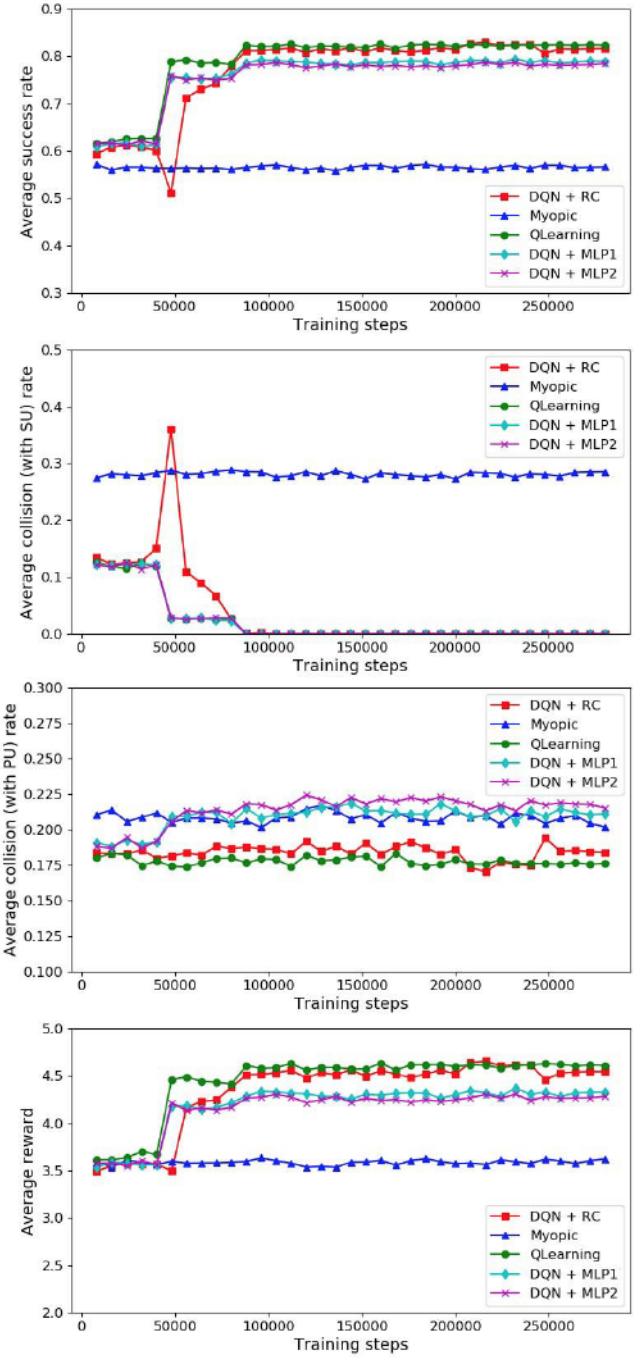


Fig. 9. Average success number, collision number with PUs, collision number with SUs, and reward with the training time (number of channel = 6 and number of SU = 2).

beginning of training because it has full knowledge of the system dynamics. However, the myopic method cannot benefit the SU from learning the access strategy of other SUs. Therefore, the myopic method will result a constant collision rate with other SUs. On the other hand, the learning-based methods are able to gradually learn from the stochastic environment and the interactions among SUs and PUs, therefore, learning-based methods all develop strategies that have higher average reward than the myopic method. As shown in Fig. 9, all learning-based methods have 0 collision among SUs when their training curves converge, meaning each SU learns the

access strategy of other SUs perfectly. The performance comparison among learning-based methods reflects the following relationship: $DQN+RC \approx Q\text{-learning} > DQN+MLP1 \approx DQN+MLP2$. To be specific, $Q\text{-learning}$ and $DQN+RC$ have similar performance because the number of channels is only 6 in this experiment. Clearly, $DQN+RC$ outperforms both $DQN+MLP1$ and $DQN+MLP2$. This is because the recurrent structure of RC enables a more complete construction of a high dimensional functional space to approximate the network state space than MLP, even though both RC and MLP have an equal number of neurons. An interesting observation is that $DQN+MLP2$ performs slightly worse than $DQN+MLP1$ even $DQN+MLP2$ has more complicated structure. This is mainly because a deeper network usually needs more training data and training time to make it converge. Therefore, having more neurons in a neural network does not always translate to having better performance.

C. Comparison Between $DQN + RC$ and $DQN + MLP1$

In this section, we demonstrate the impact of temporal correlation of the spectrum sensing outcomes on the learning performance of $DQN+MLP1$ and $DQN+RC$. Note that, we do not include $DQN+MLP2$ in this comparison since the performance of $DQN+MLP1$ is similar to $DQN+MLP2$ as discussed in Section V-B. It is important to note that in reality the spectrum sensing outcomes are correlated over time due to: 1) PUs' activities have temporal correlation and 2) the wireless channel demonstrates temporal correlation. A simple experiment is designed where the PUs' activities is time dependent. In the experiment, we assume that there is only one channel, one PU, and one SU for the underlying DSA network. The number of training sequences in each iteration is set to 2000. Instead of using the memoryless two-state Markov chain to model the PU's behavior, we assume PU's activity is time-dependent (that is, it is governed by a process which has memory). For simplicity, we assume that a PU will access the channel once every three time slots with $\{Inactive, Inactive, Active\}$ as a period.

The structure of the $MLP1$ contains one hidden layer of 64 neurons. To make a fair comparison, the number of neurons of RC is also set to 64. Besides, the learning rate of both $MLP1$ and RC is 0.01. The results are shown in Fig. 10. Because the expected reward of access the channel is higher than not accessing the channel, the strategy of $DQN+MLP1$ is to always access the channel regardless of spectrum sensing results, resulting in the success rate to be $1/3$ and the collision rate to be $2/3$. On the other hand, $DQN+RC$ could precisely learn the activity of the PU to access the channel only when the PU is *Inactive*. This is because $DQN+RC$ can effectively learn the temporal correlation of the sensing outcome, which can be used to improve access performance. The experiment result clearly demonstrates that RC is able to learn the underlying temporal correlation, while MLP can only learn the one-to-one mapping from state to action.

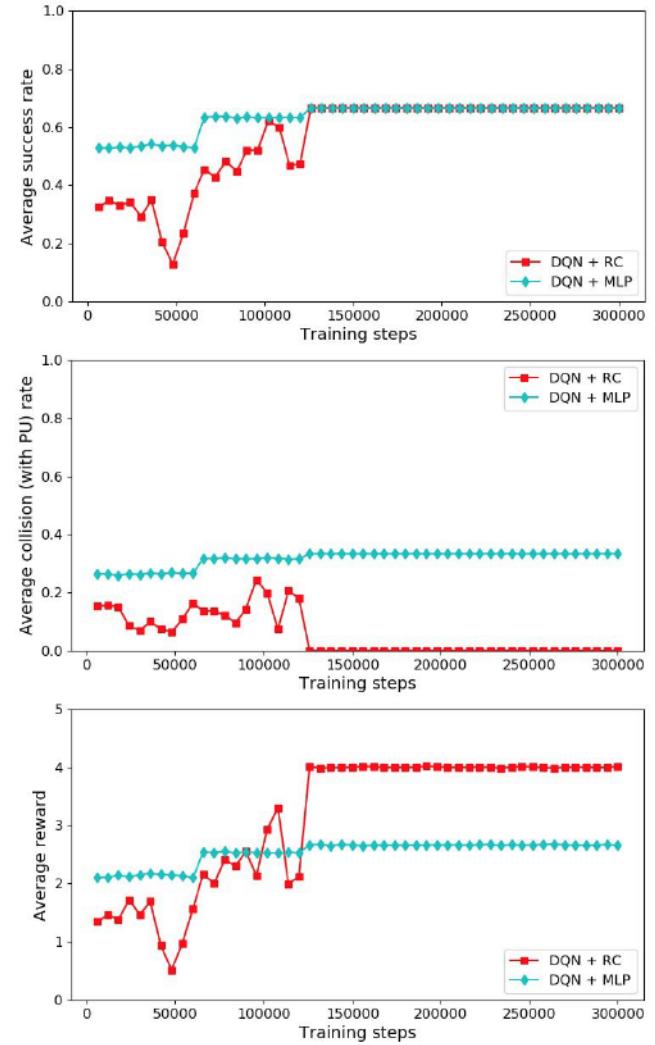


Fig. 10. Average success number, collision number with PUs, and reward with the training time (number of channel = 1 and number of SU = 1).

VI. CONCLUSION

In this paper, we study spectrum access strategies in a distributive DSA network under the condition of imperfect spectrum sensing and no centralized controllers. A new DSA strategy based on the combination of DQN and RC is introduced, in which DRL is employed to learn the availability of spectrum resources, and RC is utilized to realize DQN by taking advantage of the underlying temporal correlation of the DSA network. Equipped with our DSA strategy, each SU is able to make proper spectrum access decisions distributedly relying only on minimal broadcast information from the PUs, their own spectrum sensing outcomes, as well as the learning outcomes. Experiments are designed to verify the performance of our proposed strategy. The experiment results demonstrate that our learning method could provide higher successful transmission rate and lower transmission collision rate as oppose to the myopic strategy which generally assumes channel statistics. As compared to the $Q\text{-learning}$, our DQN-based approach is shown to converge faster with better performance. Due to the recurrent structure of the RC, our $DQN+RC$ -based scheme can

outperform DQN+MLP-based scheme when spectrum sensing outcomes exhibit temporal correlation.

REFERENCES

- [1] M. McHenry, "CISCO visual networking index: Forecast and methodology, 2016–2021," San Jose, CA, USA, Cisco, Whitepaper, Jun. 2003.
- [2] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1033–1046, Jun. 2012.
- [3] M. H. Islam *et al.*, "Spectrum survey in Singapore: Occupancy measurements and analyses," in *Proc. 3rd Int. Conf. Cogn. Radio Orient. Wireless Netw. Commun. (CrownCom)*, Singapore, 2008, pp. 1–7.
- [4] M. A. McHenry, P. A. Tenhula, D. McCloskey, D. A. Roberson, and C. S. Hood, "Chicago spectrum occupancy measurements & analysis and a long-term studies proposal," in *Proc. 1st Int. Workshop TAPAS*, Aug. 2006, p. 1.
- [5] *Shared Spectrum Company*. Accessed: Aug. 6, 2018. [Online]. Available: <http://www.sharespectrum.com>
- [6] P. J. Kolodzy, "Spectrum policy task force report," FCC, Washington, DC, USA, Rep. ET 02-135, Nov. 2002.
- [7] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.
- [8] *Federal Communications Commission*. Accessed: Aug. 6, 2018. [Online]. Available: http://transition.fcc.gov/Daily_Releases/Daily-Business/2016/db0714/DOC-340310A1.pdf
- [9] S. Bhattacharai, J.-M. J. Park, B. Gao, K. Bian, and W. Lehr, "An overview of dynamic spectrum sharing: Ongoing initiatives, challenges, and a roadmap for future research," *IEEE Trans. Cogn. Commun. Netw.*, vol. 2, no. 2, pp. 110–128, Jun. 2016.
- [10] "Report and order and second further notice of proposed rulemaking," FCC, Washington, DC, USA, Rep. GN 12-354, 2015.
- [11] Y. Liao, T. Wang, L. Song, and Z. Han, "Listen-and-talk: Full-duplex cognitive radio networks," in *Proc. GLOBECOM*, Dec. 2014, pp. 3068–3073.
- [12] Z. Fu, W. Xu, Z. Feng, X. Lin, and J. Lin, "Throughput analysis of LTE-licensed-assisted access networks with imperfect spectrum sensing," in *Proc. WCNC*, Mar. 2017, pp. 1–6.
- [13] S. Stotas and A. Nallanathan, "Enhancing the capacity of spectrum sharing cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 3768–3779, Oct. 2011.
- [14] H. Li, "Multi-agent *Q*-learning for competitive spectrum access in cognitive radio systems," in *Proc. 5th IEEE Workshop Netw. Technol. Softw. Defined Radio Netw.*, Jun. 2010, pp. 1–6.
- [15] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.
- [16] O. Napolitano and K. Cohen, "Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks," in *Proc. GLOBECOM*, Dec. 2017, pp. 1–7.
- [17] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, 2013, pp. 1310–1318.
- [18] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.
- [19] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. ISCA*, Sep. 2014, pp. 338–342.
- [20] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4694–4708, Oct. 2018.
- [21] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [22] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," German Nat. Res. Center Inf. Technol., Cologne, Germany, GMD Rep. 148, p. 13, Jan. 2001.
- [23] C. J. C. H. Watkins and P. Dayan, "*Q*-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.
- [24] E. R. Gomes and R. Kowalczyk, "Dynamic analysis of multiagent *Q*-learning with ϵ -greedy exploration," in *Proc. ICML*, Jun. 2009, pp. 369–376.
- [25] V. Mnih *et al.*, "Playing Atari with deep reinforcement learning," Deepmind Technol., London, U.K., Rep., Dec. 2013.
- [26] T. Mikolov, M. Karafiat, L. Burget, J. Černocky, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. ISCA*, Sep. 2010, pp. 1045–1048.
- [27] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
- [28] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the 'echo state network' approach," GMD-Forschungszentrum Informationstechnik Bonn, Bonn, Germany, Rep. 159, 2002.
- [29] P. Enel, E. Procyk, R. Quilodran, and P. F. Dominey, "Reservoir computing properties of neural dynamics in prefrontal cortex," *PLoS Comput. Biol.*, vol. 12, no. 6, Jun. 2016, Art. no. e1004967.
- [30] H. Jaeger, "Echo state network," *Scholarpedia*, vol. 2, no. 9, p. 2330, 2007.
- [31] K. Hamedani, L. Liu, R. Atat, J. Wu, and Y. Yi, "Reservoir computing meets smart grids: Attack detection using delayed feedback networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 734–743, Feb. 2018.
- [32] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.
- [33] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, "Winner II channel models," in *Radio Technologies and Concepts for IMT-Advanced*. Hoboken, NJ, USA: Wiley, 2009, pp. 39–92.
- [34] K. Liu, Q. Zhao, and B. Krishnamachari, "Dynamic multichannel access with imperfect channel state detection," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2795–2808, May 2010.
- [35] Y. Li, S. K. Jayaweera, M. Bkassiny, and K. A. Avery, "Optimal myopic sensing and dynamic spectrum access in cognitive radio networks with low-complexity implementations," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2412–2423, Jul. 2012.
- [36] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, "Applications of reinforcement learning to cognitive radio networks," in *Proc. ICC Workshops*, May 2010, pp. 1–6.
- [37] N. Morozs, T. Clarke, and D. Grace, "Distributed heuristically accelerated *Q*-learning for robust cognitive spectrum management in LTE cellular systems," *IEEE Trans. Mobile Comput.*, vol. 15, no. 4, pp. 817–825, Apr. 2016.



Hao-Hsuan Chang received the B.Sc. degree in electrical engineering and the M.S. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2014 and 2015, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Virginia Tech, Blacksburg, VA, USA.

His current research interests include dynamic spectrum access, deep learning, and optimization.



Hao Song (S'14) received the B.E. degree in electrical information engineering from Zhengzhou University, Zhengzhou, China, in 2011. He is currently pursuing the Ph.D. degree with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA.

His current research interests include mobile network optimization, dynamic spectrum access, and machine learning and its applications in wireless communications.



Yang Yi (SM'17) received the B.S. and M.S. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA.

She is an Assistant Professor with the Bradley Department of ECE, Virginia Tech, Blacksburg, VA, USA. Her current research interests include very large scale integrated circuits and systems, computer-aided design, and neuromorphic computing.

Dr. Yi was a recipient of the 2018 National Science CAREER Award, the 2016 Miller Professional Development Award for Distinguished Research, the 2016 United States Air Force Summer Faculty Fellowship, the 2015 NSF EPSCoR First Award, and the 2015 Miller Scholar. She is currently serving as an Associate Editor for Cyber Journals: *Journal of Selected Areas in Microelectronics* and has been serving on the editorial board of the *International Journal of Computational & Neural Engineering*.



Haibo He (SM'11–F'18) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, OH, USA, in 2006.

From 2006 to 2009, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. He is currently the Robert Haas Endowed Chair Professor with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, South Kingstown, RI, USA. He has published one sole-author research book (Wiley), edited one book (Wiley-IEEE), and six conference proceedings (Springer), and has authored and co-authored over 300 peer-reviewed journal and conference papers. He has delivered over 100 invited talks around the globe. His current research interests include adaptive learning and control, computational intelligence, machine learning, data mining, and various applications.

Dr. He was a recipient of the IEEE International Conference on Communications Best Paper Award in 2014, the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2014, and the National Science Foundation CAREER Award in 2011. He was the Chair of IEEE CIS Emergent Technologies Technical Committee in 2015 and the IEEE CIS Neural Networks Technical Committee in 2013 and 2014. He served as the General Chair of 2014 IEEE Symposium Series on Computational Intelligence, Orlando, FL, USA. He is currently the Editor-in-Chief of the *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*.



Jianzhong (Charlie) Zhang (F'16) received the Ph.D. degree from the University of Wisconsin–Madison, Madison, WI, USA.

He is a VP and the Head of the Standards and Mobility Innovation Laboratory, Samsung Research America, Richardson, TX, USA, where he leads research, prototyping, and standards for 5G and future multimedia networks. From 2009 to 2013, he served as the Vice Chairman of the 3GPP RAN1 working group and led development of LTE and LTE-advanced technologies, such as 3-D channel modeling, UL-MIMO, CoMP, and carrier aggregation for TD-LTE.



Lingjia Liu (SM'15) received the B.S. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA.

He was an Associate Professor with the EECS Department, University of Kansas (KU), Lawrence, KS, USA. He was with the ECE Department, Virginia Tech, Blacksburg, VA, USA. He spent over four years with Mitsubishi Electric Research Laboratory, Cambridge, MA, USA, and the Standards and Mobility Innovation Laboratory, Samsung Research America, Richardson, TX, USA, where he received Global Samsung Best Paper Award in 2008 and 2010. He was leading Samsung's efforts on multiuser MIMO, CoMP, and HetNets in LTE/LTE-Advanced standards. His current research interests include emerging technologies for 5G cellular networks, including machine learning for wireless networks, massive MIMO, massive MTC communications, and mmWave communications.

Dr. Liu was a recipient of the Air Force Summer Faculty Fellow from 2013 to 2017, the Miller Scholar at KU in 2014, the Miller Professional Development Award for Distinguished Research with KU in 2015, the 2016 IEEE GLOBECOM Best Paper Award, the 2018 IEEE ISQED Best Paper Award, and the 2018 IEEE TAOS Best Paper Award.