Robust Deep Reinforcement Learning for Interference Avoidance in Wideband Spectrum

Mohamed A. Aref
Communications and Information Sciences Laboratory
ECE Department, University of New Mexico
Albuquerque, NM, USA
maref@unm.edu

Sudharman K. Jayaweera
Communications and Information Sciences Laboratory
ECE Department, University of New Mexico
Albuquerque, NM, USA
jayaweera@unm.edu

***Abstract*—This paper presents a design of a cognitive engine for interference and jamming resilience based on deep reinforcement learning (DRL). The proposed scheme is aimed at finding the spectrum opportunities in a heterogeneous wideband spectrum. In this paper we discuss a specific DRL mechanism based on double deep Q-learning (DDQN) with a convolutional neural network (CNN) to successfully learn such interference avoidance operation over a wideband partially observable environment. It is shown, through simulations, that the proposed technique has a low computational complexity and significantly outperforms other techniques in the literature, including other DRL-based approaches.**

***Index Terms*—Convolutional neural network, deep reinforcement learning, double deep Q-network, interference avoidance, wideband autonomous cognitive radio.**

I. Introduction

Deep learning (DL) is expected to play an important role in future wireless communications networks design in many fields including space, military and consumer wireless communications [1]–[3]. Recently, Deep reinforcement learning (DRL), a branch of DL, has been proposed for several applications in wireless communications that require autonomous decision-making, including power control and network access [3]. Another important application is network reliability and security in which the radio adapts to avoid jamming and other malicious attacks [4]–[7].

The authors in [4] proposed a two dimensional anti-jamming technique using a standard deep Q-network (DQN) with convolutional neural network (CNN). At any time instant, the radio performs two actions: (1) Stay in or leave the current cell (2) Select a frequency channel for communications. In [5], the problem of anti-jamming transmission in unmanned aerial vehicle (UAV) systems is discussed. More specifically, the authors discussed an optimal power allocation strategy using DQN to resist smart attacks. In [6], the authors considered the same problem formulation as in [4] in which the radio attempts to learn an efficient frequency hopping policy. The proposed anti-jamming technique is based on DQN with a recursive convolutional neural network (RCNN). Spectrum waterfall that includes both temporal and spectral information was used as the state of the DQN in [6]. Most of the above referenced contributions, however, either do not have the ability to work in a wideband partially-observable spectrum or suffer from high computational complexity.

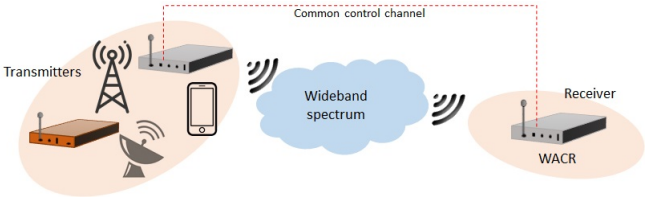

Figure 1. System model including heterogeneous RF environment.

Most recently, a promising cognitive interference avoidance technique is proposed in [7] that is suitable for real-time applications in partially-observable spectrum. The proposed scheme is based on DRL, more specifically, double deep Q-network (DDQN) with CNN [8], [9]. It is assumed that the radio can sense the surrounding RF spectrum while maintaining the communications link of interest. The information collected from both sensing and communications is used to create the state of the DDQN algorithm. The radio attempts to learn an effective policy to choose a frequency channel with the highest SINR for communications at every time instant. Although, the proposed technique outperforms many of the traditional techniques including Q-learning and DQN, the simulation results showed that there is still a room for improvement.

The purpose of this paper is to extend the framework of [7] to improve the performance and reduce the computational complexity. First, we developed new definitions for state and operation parameters. Second, we simplified the CNN architecture used in the DDQN. Finally, we povided a detailed description of the proposed DDQN including the interactions between the sensing and communications operations. One of the advantages of the proposed cognitive engine is compatibility with both high and low performance computing platforms due to the simplicity of the state definition and the CNN architecture.

The remainder of this paper is organized as follows: Section II introduces the system model. The details of the proposed cognitive engine design based on DDQN are explained in Section III. Section IV presents the simulation results. Discussion

and conclusions are described in section V.

II. System Model

We assume that a wideband autonomous cognitive radio (WACR) represents the receiver of our link of interest [7], [10]. The objective of the WACR is to preserve the connectivity by choosing a frequency channel with highest SINR for communications at every time instant. The surrounding RF environment includes multiple interference and jamming signals as shown in Fig. 1. The frequency synchronization between the receiver and the transmitter is done through a common control channel [7]. The RF spectrum of interest is assumed to have N non-overlapping channels.

The WACR performs two operations simultaneously: communications and sensing. For communications, the WACR chooses an action at each time instant t, denoted by $a^c(t) \in \mathcal{A}^c$, that represents the index of the channel for communications at time instant $t+1$. The action set of the communications operation is denoted by $\mathcal{A}^c = \{1, \cdots, N\}$. The transmitter sends the signal of interest with a given power P_s. The channel power gain from the transmitter to the WACR is given by h_s. On the same channel, the interference source i and the jammer j transmit their signals with given powers $P_{I,i}$ and $P_{J,j}$. On the other hand, channel power gains to the WACR are $h_{I,i}$ and $h_{J,j}$, from sources i and j, respectively. The received SINR of the WACR in channel $a^c(t)$ at time t can be expressed as

$$\mu_{a^c(t)} = \frac{h_s P_s}{\sigma^2 + \sum_i h_{I,i} P_{I,i} + \sum_j h_{J,j} P_{J,j}}, \tag{1}$$

where σ^2 is the receiver noise power. Let μ_{th} denotes the required SINR threshold for successful communications. Then, a function $g(.)$ that indicates the success of the communications over channel $a^c(t)$ at time t can be defined as follows:

$$g(\mu_{a^c(t)}) = \begin{cases} \lambda & \text{if } \mu_{a^c(t)} > \mu_{th} \text{ (success)} \\ -\lambda & \text{if } \mu_{a^c(t)} \leq \mu_{th} \text{ (failure)}, \end{cases} \tag{2}$$

where $\lambda > 0$ is a design parameter selected to provide sufficient contrast between the two cases for efficient learning.

For sensing, the WACR senses N_s channels at a time, where $N_s \leq N$ due to hardware constraints. Assume that at time t, the action for sensing operation is $\mathbf{a}^s(t) = [a_1^s(t), \cdots, a_{N_s}^s(t)]$ that represents the set of sensing channel indices. At time t, the WACR can estimate the power spectral density (PSD) $\nu_{a_i^s(t)}$ for the sensed channel $a_i^s(t) \in \mathbf{a}^s(t)$. Using spectral activity detection, the WACR can then identify the availability of channel by comparing $\nu_{a_i^s(t)}$ with an appropriate threshold ν_{th} that is designed based on noise floor estimation [10]. Similar to (2), let function $f(\nu_{a_i^s(t)}) = -\lambda$ denotes the unavailability of the channel for $\nu_{a_i^s(t)} > \nu_{th}$, otherwise $f(\nu_{a_i^s(t)}) = \lambda$.

Using the information from both communications and sensing, we can create a matrix $\mathbf{I}(t)$ at time t as follows:

$$\mathbf{I}(t) = \begin{bmatrix} a^c(t-1) & g(\mu_{a^c(t-1)}) \\ a_1^s(t-1) & f(\nu_{a_1^s(t-1)}) \\ \vdots & \vdots \\ a_{N_s}^s(t-1) & f(\nu_{a_{N_s}^s(t-1)}) \end{bmatrix}. \tag{3}$$

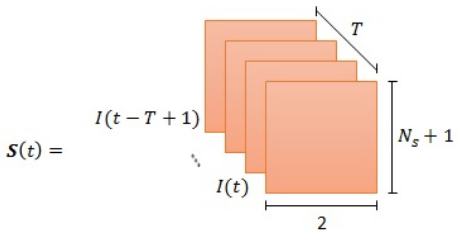

Figure 2. Proposed three dimensional state.

The matrix $\mathbf{I}(t)$ indicates the the communications success/failure as well as the availability of the sensed channels and we call it *indication matrix*. At any time t, the state $\mathbf{S}(t)$ is made of T successive indication matrices up to time t as shown in Fig. 2, where T is the temporal memory depth.

III. Proposed Cognitive Engine for Interference Avoidance using DDQN

The interference avoidance problem can be modeled as a Markov decision process (MDP) in which the WACR selects actions $a^c(t)$ and $\mathbf{a}^s(t)$ at time t. The WACR then moves from current state $\mathbf{S}(t)$ to a new one $\mathbf{S}(t+1)$ and receives a reward $r(\mathbf{S}(t), a^c(t))$. Since the objective of the WACR is to choose the communications channel with the highest SINR, the reward function is set equal to the received SINR $\mu_{a^c(t)}$. The multi-dimensional state and large state-action space make DRL a suitable learning candidate for this problem.

In this work, we proposed using DDQN with a CNN that is used to estimate the Q-values from the state $\mathbf{S}(t)$. This helps to improve the training and learning process especially when dealing with three dimensional state as in our case. Fig. 3 shows the proposed cognitive engine design using DDQN.

A. Description of the Proposed DDQN Algorithm

The WACR starts with selecting N_s channels for sensing at time $t-1$: $a_1^s(t-1), \cdots, a_{N_s}^s(t-1)$. This selection process is based on the sensing strategy that the WACR adopts (e.g. random or sweeping) [7]. At time t, the WACR can estimate the PSD in each sensing channel and by comparing with the threshold ν_{th} it can detect the availability/unavailability of the channels using the functions $f(\nu_{a_1^s(t-1)}), \cdots, f(\nu_{a_{N_s}^s(t-1)})$. On the other hand, the WACR detects the success/failure of the communications over the channel of interest by estimating the function $g(\mu_{a^c(t-1)})$.

Information from both sensing and communications are used to create the indication matrix $\mathbf{I}(t)$ at time t as shown in Fig. 3. The current matrix $\mathbf{I}(t)$ along with those from time $t-1$ to $t-T+1$ are used to create the state $\mathbf{S}(t)$ of the DDQN. A Q-network (CNN) with weights $\theta(t)$ uses $\mathbf{S}(t)$ as an input to estimate the Q-values for all possible actions in set $\mathcal{A}^c$. The WACR selects an action $a^c(t)$ that represents the index of the communications channel at time $t+1$ based on the maximum

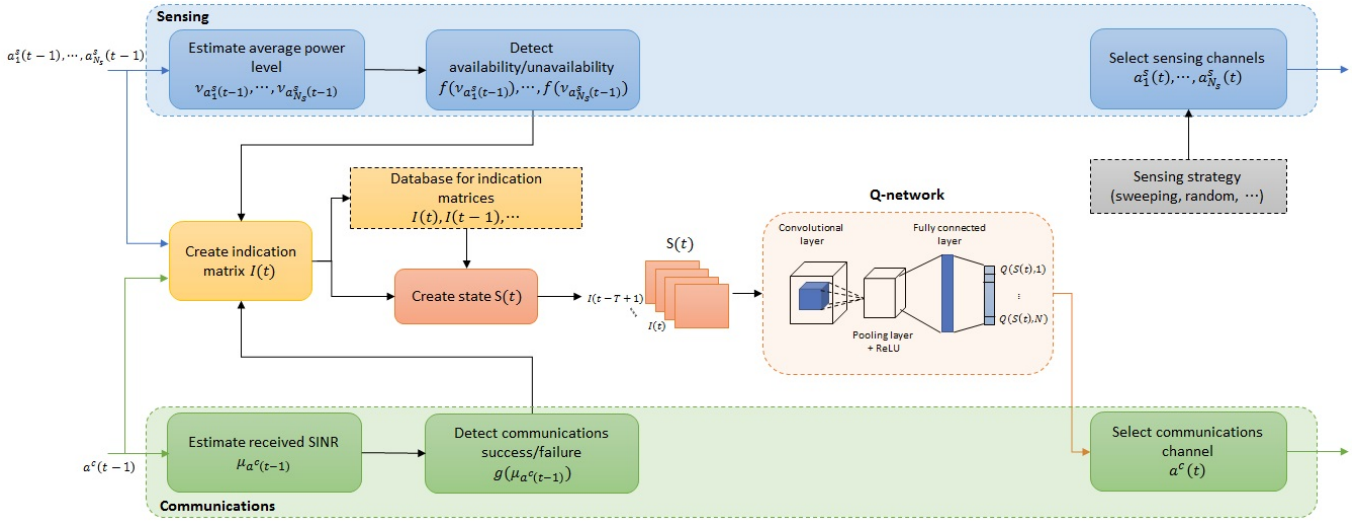

Figure 3. Proposed cognitive engine for interference avoidance and anti-jamming using DDQN.

Q-value with probability $1-\epsilon$, otherwise it randomly selects an action out of all possible actions:

$$a^c(t) = \begin{cases} \arg\max_{\acute{a}\in\mathcal{A}^c} Q(\mathbf{S}(t), \acute{a}; \theta(t)) & \text{with probability } 1-\epsilon \\ \sim U(\mathcal{A}^c) & \text{with probability } \epsilon, \end{cases} \quad (4)$$

where $U(\mathcal{A}^c)$ denotes the uniform distribution over the action set $\mathcal{A}^c$ and ϵ is an exploration rate that allows the learning algorithm to explore the space of states and actions to avoid being trapped in a sub-optimal policy.

B. Training DDQN

For DDQN training, experience replay is used in which we store WACR's experiences $x(t) = (\mathbf{S}(t), a^c(t), \mu_{a^c(t)}, \mathbf{S}(t+1))$ at each time instant t in a data set $\mathcal{D}(t) = \{x(1), \cdots, x(t)\}$. During learning at time t, we draw uniformly at random an experience $x(k) \sim U(\mathcal{D}(t))$, with $1 \leq k \leq t$, from the set of the stored experiences. The weights $\theta(t)$ of the Q-network at time t are then updated according to the stochastic gradient descent (SGD) using the loss function below:

$$L(\theta(t)) = \mathbb{E}_{x(k)\sim U(D(t))}[(\eta - Q(\mathbf{S}(t), a^c(t); \theta(t))^2], \quad (5)$$

where η is the target Q-value. This process can be repeated for K times at each time instant t in which $\theta(t)$ is updated according to K randomly selected experiences.

It is known that using the same weights $\theta(t)$ to estimate both the Q-value $Q(\mathbf{S}(t), a^c(t))$ and the target η may lead to large oscillations in the training process because at every training step when the Q-value shifts, the target value also shifts [8]. In other words, the network is trying to chase a moving target. To avoid this problem a separate Q-network, named target Q-network with weights $\hat{\theta}(t)$, is used to estimate the target value η as follows:

$$\eta = \mu_{a^c(t)} + \gamma\, Q(\mathbf{S}(t+1), a^*; \hat{\theta}(t)). \quad (6)$$

In contrast to the original Q-network, the target Q-network does not update its weights $\hat{\theta}(t)$ at every training step. Instead, the weights $\hat{\theta}(t)$ are set equal to $\theta(t)$ for every fixed number of iterations L. Thus, the target value will be constant for L successive iterations.

In DDQN, estimating the Q-value $Q(\mathbf{S}(t+1), a^*; \hat{\theta}(t))$ and selecting the best action a^* are separated as shown in (6) to avoid producing overestimated values that may degrade the learning performance and the convergence rate [9]. Thus, instead of using the target Q-network (with weights $\hat{\theta}(t)$), the Q-network (with weights $\theta(t)$) is used to estimate a^* at time t in (6) such that $a^* = \arg\max_{a'\in\mathcal{A}^c} Q(\mathbf{S}(t+1), a'; \theta(t))$.

C. Computational Complexity

As shown in Fig. 3, the proposed CNN architecture for the DDQN consists of one convolutional layer followed by average pooling layer, rectified linear units (ReLUs) and one fully connected layer. The convolutional layer includes M_1 filters with size $F_1^w \times F_1^l$, zero padding and stride 1. The fully connected layer has N rectified linear units to output the Q-value estimates for each possible action.

According to [11] the pooling and the fully connected layers often take only 5–10% of the computational time. The computational complexity of the CNN can be then approximated as the number of multiplications in the convolutional layers as follows [11]:

$$\#\, multip. = \sum_{d=1}^{D} M_{d-1} F_d^w F_d^l M_d Z_d^w Z_d^l, \quad (7)$$

where d is the index of the convolutional layer and D is the number of convolutional layers. M_{d-1} and M_d are the number of input channels and the number of filters for dth layer, respectively. F_d^w and F_d^l denote the width and length of each filter in the dth layer, respectively. Z_d^w and Z_d^l are the

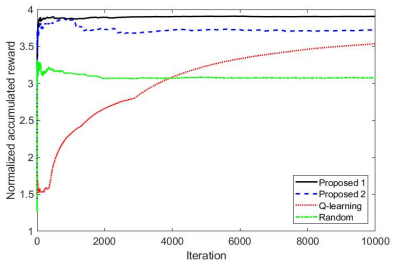

Figure 4. Normalized accumulated reward (SINR) for test case 1.

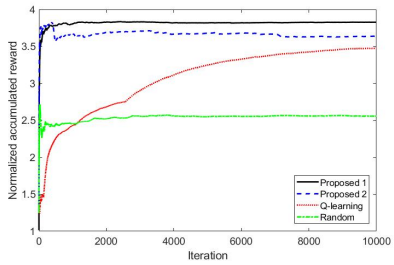

Figure 5. Normalized accumulated reward (SINR) for test case 2.

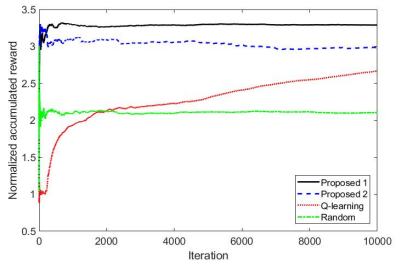

Figure 6. Normalized accumulated reward (SINR) for test case 3.

width and length of the output feature map. Z_d^l (the same for Z_d^w) is computed as [12]:

$$Z_d^l = \frac{Z_{d-1}^l - F_d^l + 2P_d}{S_d} + 1, \quad (8)$$

where P_d and S_d denote the padding and the stride at the dth layer. By applying the proposed CNN configurations in (7) and (8), the computational complexity of the proposed DDQN is given by

$$\#\, multip. = TF_1^w F_1^l M_1 (3 - F_1^w)(N_s + 2 - F_1^l), \quad (9)$$

where $Z_0^w \times Z_0^l \times M_0 = 2 \times (N_s + 1) \times T$ denotes the size of the input state as shown in Fig. 2.

IV. Simulation Results

Let us assume the signal of interest is transmitted with power $P_s = 5\ mW$, while the channel power gain to the WACR is $h_s = 0.8$. Hence, the optimal SINR value at any channel is 4 which corresponds to WACR selecting a channel free of interference and jamming. On the other hand, jamming signal j is transmitted with power $P_{J,j} = 8\ mW$ with a channel power gain to the WACR $h_{J,j} = 0.7$. Interference signal i has a transmit power of $P_{I,i}$ and channel power gain $h_{I,i}$ that take random values from the sets $[3, 6]$ and $[0.4, 0.9]$, respectively.

In all the following simulations our proposed technique is named "Proposed 1". The operation parameters are set as follows: $N = 6$, $T = 3$, $N_s = 2$, $K = 5$, $L = 10$, $\epsilon = 0.1$, $\gamma = 0.4$, $\lambda = 10$, $\sigma^2 = 1mW$, $\nu_{th} = 2$, $\mu_{th} = 2$ and learning rate of 0.1. With these parameter values, the state $\mathbf{S}(t)$ at any time t is a three-dimensional array with size $3 \times 2 \times 3$ that represents the input to the CNN. The proposed CNN architecture for the DDQN algorithm consists of one convolutional layer includes 10 filters with size 2×2 and stride 1. This is followed by average pooling layer of size 2×1 and ReLUs. Finally, a fully connected layer with 6 rectified linear units is used.

Three different techniques are used for comparison purpose to evaluate our proposed technique:

1) Proposed 2: This is the proposed technique in [7]. The state consists of sensing matrix, index of communications channel and an indicator for success/failure of the communications over this channel as described in [7]. Using the same operation parameters described above the state is then 4×6 matrix.
2) Q-learning: The Q-learning uses a simplified version of the original proposed state that includes only the index of the communications channel and the indicator for successful communications.
3) Random: The WACR randomly chooses a channel.

Table I
Performance comparison: normalized accumulated reward values after 10,000 iterations.

Test case	Scenario	Proposed 1	Proposed 2 [7]	Q-learning	Random	Optimal
1	2 inter. signals	3.9	3.7	3.5	3.1	4
2	3 inter. signals	3.8	3.6	3.4	2.5	4
3	3 inter. signals and Markov jammer	3.3	3	2.7	2.1	4

Three test cases are considered with different interference and jamming signal scenarios. Table I shows the performance comparison in terms of normalized accumulated reward after 10,000 iterations with a scenario description for each test case. Test case 1 represents a simplified scenario in which there are only two interference sources that transmit continuously their signals over two dedicated channels. Fig. 4 shows the normalized accumulated reward for this scenario. The proposed technique achieves about 97.5% of the maximum possible reward, while Proposed 2 and Q-learning achieve 92.5% and 87.5%, respectively.

In test case 2, besides the two interference sources in test case 1, an extra interference source is added that switches between ON and OFF in a random manner. From Table I and Fig. 5, the proposed technique (Proposed 1) significantly outperforms Proposed 2, Q-learning and random in terms of the accumulated reward. In test case 3, there is a Markov jammer operating besides the 3 interference signals described in test case 2 [7]. Fig. 6 shows the normalized accumulated reward for this scenario. Similar to the previous two test cases, the proposed technique shows an improvement in the performance when compared with other techniques.

Table II summarizes the configurations of the CNN models in Proposed 1 and Proposed 2 algorithms. The "Comp." is the theoretical computational complexity relative to Proposed

Table II
CNN PARAMETERS OF THE PROPOSED ALGORITHMS.

	Input	Conv. 1	Conv. 2	Pool	FC	Comp.
Proposed 1	$3 \times 2 \times 3$	$2 \times 2 \times 10$	/	2×1	6	0.0196
Proposed 2 [7]	$4 \times 6 \times 1$	$1 \times 1 \times 10$	$2 \times 2 \times 20$	/	6	1

2 using (7). It is clear from Table II that using the Proposed 1 algorithm can significantly reduce the computational complexity by 98% compared with Proposed 2.

V. CONCLUSION

In this paper we have presented a cognitive interference avoidance and anti-jamming scheme based on DRL. In particular a DDQN with a simplified CNN is used to learn an efficient policy to a void harmful signals and maintain a communications link of interest in a heterogeneous partially-observable spectrum. The state of the DDQN is a three dimensional array that indicates the availability of the sensed channels as well as the success/failure of communications over the link of interest. Results obtained from simulation showed that the proposed technique can learn an effective policy to avoid interference and jamming signals. Furthermore, it significantly outperforms similar techniques from literature while reducing the computational complexity.

ACKNOWLEDGMENT

This research was sponsored in part by the Army Research Laboratory and was accomplished under Grant Number W911NF-17-1-0035. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] S. Dorner, S. Cammerer, J. Hoydis and S. Brink, "Deep Learning based Communication Over the Air," IEEE Journal of Selected Topics in Signal Processing, Vol. 12, no. 1, Feb. 2018.

[2] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilé, R. C. Reinhart and D. J. Mortensen, "Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles," IEEE Journal on Selected Areas in Communications, vol. 36, no. 5, May 2018.

[3] N. Luong, D. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. Kim, "Applications of deep reinforcement learning in communications and networking: a survey," arXiv, eprint arXiv:1810.07862 [cs.NI], 2018.

[4] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, Mar. 2017.

[5] L. Xiao, C. Xie, M. Min, and W. Zhuang, "User-ventric view of unmanned aerial vehicle transmission against smart attacks," IEEE Transactions on Vehicular Technology, vol. 67, no. 4, Apr. 2018.

[6] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," IEEE Communications Letters, vol. 22, no. 5, May 2018.

[7] M. A. Aref and S. K. Jayaweera, "Spectrum-agile cognitive interference avoidance through deep reinforcement learning," 14th EAI International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM'19), Poznan, Poland, Jun. 2019.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, Jan. 2015.

[9] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Phoenix, AZ, USA, Feb. 2016.

[10] S. K. Jayaweera, "Signal processing for cognitive radios," 1st ed. New York, NY, USA: John Wiley & Sons Inc., 2014.

[11] K. He and J. Sun, "Convolutional neural networks at constrained time cost," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, Jun. 2015.

[12] F. Li, J. Johnson and S. Yeung, "Stanford University CS231n: Convolutional Neural Networks for Visual Recognition," http://cs231n.github.io/convolutional-networks/, accessed on 5th Apr. 2019.