Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

An overview of deep reinforcement learning for spectrum sensing in cognitive radio networks

Felix Obite [a,*], Aliyu D. Usman [b], Emmanuel Okafor [c]

[a] Department of Physics, Faculty of Physical Science, Ahmadu Bello University, P.M.B. 1044, Zaria, Nigeria
[b] Department of Electronics and Telecommunications Engineering, Ahmadu Bello University, P.M.B. 1044, Zaria, Nigeria
[c] Department of Computer Engineering, Ahmadu Bello University, P.M.B. 1044, Zaria, Nigeria

ARTICLE INFO

Article history:
Available online 5 March 2021

Keywords:
Deep reinforcement learning
Cognitive radio networks
Spectrum sensing
Markov decision process
Cooperative sensing
Spectrum holes

ABSTRACT

Deep reinforcement learning has recorded remarkable performance in diverse application areas of artificial intelligence: pattern recognition, robotics, object segmentation, recommendation-system, and gaming. In recent times, the applicability of deep learning to telecommunication technology is gradually attracting a lot of attention, especially in spectrum sensing, a core component in cognitive radio. The traditional approaches to spectrum sensing are heavily prone to noise uncertainty and often rely on either complete or partial prior knowledge of the primary users. An alternative method that can curb the aforementioned problem is deep reinforcement learning, which integrates several layers of neural networks for extracting and learning features automatically from a given data. Hence, we survey and propose a theoretical hypothetic model formulation of deep reinforcement learning as an effective method for creating a cooperative spectrum sensing model that can overcome the limitations of traditional spectrum sensing methods, which are often prone to low sensing precision. Also, the study provides an overview of past, current, and future advances in cognitive radio networks. The discussion herein will be of interest to a wide range of audiences in telecommunication and artificial intelligence.

1. Introduction

Cognitive radio (CR) is the main technology enabling future wireless networks, to exploit the radio spectrum more efficiently and opportunistically without causing interference to the primary users (PUs). It is a radio having the ability to alter its transmitter features by interacting with its immediate environment [1]. CR is different from traditional radio systems since it has the advantage of equipping users with both cognitive ability and reconfigurability [2,3]. A cognitive ability entails the capability of sensing and gathering relevant information from the immediate environment. With this ability, a secondary user (SU) can detect the best accessible spectrum. Re-configurability is the ability to adapt the operating features speedily based on the sensed signal to attain optimum performance. By utilizing the spectrum opportunistically, CR permits SUs to scan the portion of the radio spectrum that is free, select the best channel that is free, establish spectrum access, then relinquish the channel to the licensed PU when the PU becomes active. By permitting a better flexible utilization of the radio spectrum, particularly when SUs are coexisting with PUs, conventional spectrum allocation patterns [4] with spectrum access strategies will not be required anymore. Novel spectrum allocation strategies are required to resolve CR-related research, specifically, spectrum sensing, also known as dynamic spectrum access (DSA). Moreover, it is quite difficult and costly to obtain extra spectrum bands for next-generation wireless systems. Thus, improving spectrum utilization is crucial for the attainment of 5G and future wireless technology which has the prowess to process and transport a large volume of data within a minimum time delay.

The cognitive framework initially was defined by Mitola in the context of seven basic functionalities as illustrated in Fig. 1 [5]. The seven functionalities stand out in this definition: learning, observation, orienting, planning, deciding, action, and autonomy. In the research paper by Haykin [2] the cognitive cycle is described in terms of communications, learning capabilities, and signal processing concepts. The cognitive framework as described by Mitola [21] has been reframed concerning dynamic spectrum management [6]. Presently, the active research community has acknowledged several cognitive tasks that are susceptible to further studies [7], [8]. Similarly, many CR test-beds from a real viewpoint have been effectively developed [9], [10].

Firstly, the sensing task is performed to learn the radio space. The second functionality is observation, which is related to assessing the environment using the sensed information or signal. The third functionality is orienting; for instance, to familiarize itself to

[*] Corresponding author.
E-mail address: felixobite@gmail.com (F. Obite).

https://doi.org/10.1016/j.dsp.2021.103014

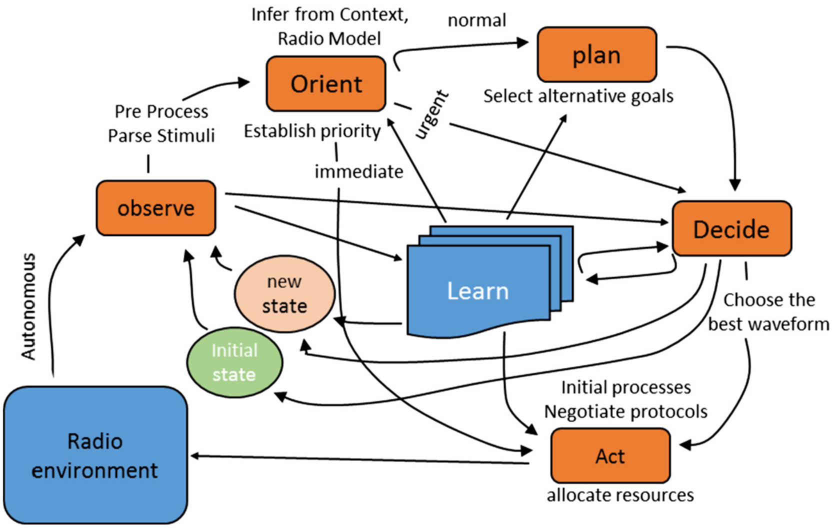

Fig. 1. CR concept by Mitola [5].

choose which action to execute. The other functionality includes planning which identifies other actions, drawing conclusions that determine feasible action, applying action to accomplish the task, and finally, earning autonomy through experience gained from the earlier six functionalities [11].

Usually, three basic models exist for CR networks, they include; the interweave CR, the underlay CR, and the overlay CR. Firstly, for the interweave CR, unlicensed or CR users are restricted from accessing the occupied PU band. In such a scenario, the CR user will have to detect underutilized spectrum holes at a specific time interval and a particular geographic region. Hence, the major task confronting a CR node is sensing the radio spectrum to ascertain if the licensed PU is active or not. Consequently, the key motivation for the interweave CR model is SS. Given that, SS [12] could be defined as the process of identifying the white spaces or spectrum holes as depicted in Fig. 2. Also, the CR user is expected to immediately leave the channel as soon as the PU becomes active to minimize harmful interference to the licensed PU. SS techniques may be classified based on wideband or narrowband. Wideband SS [13] involves monitoring a wideband to identify the portions that are fully occupied or free. In contrast, narrowband SS involves monitoring only a single portion of such band [14]. Secondly, for the underlay CR, PUs and SUs are permitted to coexist together and thus the model is also called a spectrum sharing model [15–18]. Nevertheless, the licensed PUs are usually given the highest priority over the CR users to occupy the spectrum. Also, the coexistence is defined within the PU's interference limit (i.e., based on a specified interference threshold). Normally, the CR user will spread its signal around a very large bandwidth that can guarantee a minimum interference level to the PU. As a result of this limit, the underlay model is only suitable for short-range networks. Thirdly, in overlay CR, the PUs and CR users are permitted to transmit simultaneously. The key assumption supporting the overlay CR model has based on the premise that the CR user has prior knowledge of the PU [18]. Two basic techniques are employed for this model. First, by utilizing advanced coding methods [19] for example dirty paper coding, where the CR user may pre-code the transmitted signal to efficiently nullify the interference level at the CR receiver. This method offers a theoretic upper limit on the highest attainable throughput by the CR users. Second, the CR user power is divided into two parts, one part is utilized to increase the PU power to reduce the effect of the interference produced by the CR user data, while the second part is used to convey the CR user data [20].

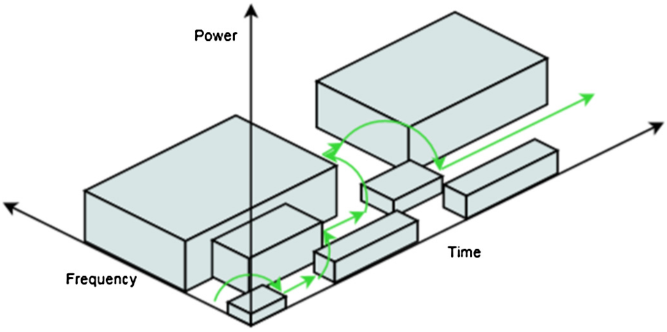

Fig. 2. Spectrum holes or white spaces in time and frequency domain. The grey blocks indicate the (PUs) while the spectrum holes or white spaces are shown by the green arrow [21]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Lately, deep learning (DL) [22] an advanced form of machine learning is aided to resolve traditional RL limitations. Moreover, DL has evolved to a new age that advances RL to deep reinforcement learning (Deep RL). Dealing with data and computation, Deep RL offers a generic and flexible paradigm for evaluating complex problems that perfectly match the requirement of spectrum management in cognitive radio systems.

Despite the increasing interest in deep learning (DL) and RL in the CR domain, a comprehensive overview focusing on Deep RL for SS in CR is lacking. To facilitate the application of Deep RL for SS in CR, we present a complete overview of the advanced research in this field. This article is intended to fill the gap between Deep RL and SS in CR, by presenting a state-of-the-art review for interested practitioners to further advance this field.

1.1. Motivation and main contributions

The increasing need for extra bandwidth to assist new data-intensive applications has triggered the present spectrum scarcity which often translates to inefficient usage of spectrum resources. Deep RL offers a generic and flexible paradigm for evaluating complex problems that perfectly match the requirement of SS in CR networks. Thus, the novel contributions for this review are highlighted as follows.

- We present a complete overview of the advanced research within the domain of Deep RL for SS in CR. This is the first

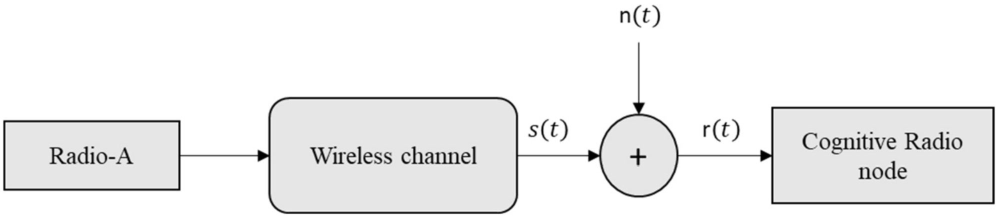

Fig. 3. System model for energy detection (ED) [35].

overview of the best of our knowledge that is focusing on SS in CR using Deep RL.

- We propose a mathematical hypothetic model of a Deep RL-based cooperative spectrum sensing model in the context of a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). Furthermore, we propose that any of the Deep RL algorithms discussed can be used to formulate the optimal solution. Since deep RL has the potential to learn features automatically from data within a given environment without previous knowledge of the underlying distributions, the proposed model is envisaged to outperform the traditional spectrum sensing techniques.
- We present insights into the challenges for the implementation of Deep RL for SS in CR and identify future research directions that remain unresolved and are worth exploring with Deep RL.

1.2. Paper outline

The remaining components of the paper are highlighted as follows: Section 2 discusses the SS techniques, which entails how traditional SS issues are solved, fundamentals of spectrum sensing and detection, and a review of related work. Section 3 provides a theoretical background necessary for understanding Deep RL techniques. Section 4 presents an overview of Deep RL for spectrum sensing in CR systems. Potential challenges for spectrum sensing with deep RL are presented in Section 5. The conclusion and future research are highlighted in Section 6.

2. Spectrum sensing techniques

Spectrum sensing techniques could be classified into 2 main classes: non-cooperative sensing and cooperative sensing (Fig. 3). Non-cooperative sensing techniques involve detecting transmitted signals from a PU based on local observations from CRs. Non-cooperative sensing approaches are built on the notion that the position of the PU is not known to the CR node. As a result, the CR users rely on weak PU signals for SS. A CR node can only have partial knowledge of the PU activity. Therefore, it is unable to fully prevent destructive interference to PUs. Furthermore, non-cooperative sensing cannot avoid the hidden node problem in CR. Three main techniques are typically used for non-cooperative detection: energy detection matched filter detection and features detection. As a result of deep fading and shadowing, a CR user may not be able to detect the hidden node problem. Thus, cooperative sensing schemes are employed to alleviate this shortcoming. Cooperative sensing involves several CR nodes sharing their local observations for improved PU detection [23–26]. Cooperative sensing could be realized in a centralized or distributed fashion. With the centralized pattern, a central station gathers sensing results from CR users, detects a free channel, and broadcasts the result to other CR users [23]. With the distributed method, a central station is not required, but rather, the sensing result is shared with other CR users in a distributed or sequential manner [23]. Cooperative sensing approaches can also be categorized into soft or hard sensing schemes. Hence there is a need to understand the difference between these two methods. The soft and hard cooperative sensing can be defined in the context of their combination and decision schemes, based on how the sensing result is shared among CR users. The soft combination is a cooperative scheme where a node senses a specific frequency and transfers the information to a central station [27–30]. At the central station, a final decision is made on the presence of the PU by fusing all the information received. On the other hand, in a hard combination scheme, a node decides if a PU is active (i.e., where bit-1 and bit-0 denote the presence and absence of the PU respectively) then transfers the result of the decision to a central station. At the central station, the 1-bit decision from all the different nodes is fused to give a global decision of the presence of the PU [31]. The performance of the soft combination scheme is higher but at the expense of higher bandwidth constraints for the control channel. Hard strategy, though less precise, entails less sharing of information among other nodes. Similarly, with distributed sensing, every contributing node executes its fusion policy locally using the information from other nodes in the system.

2.1. Fundamentals of SS and detection

The CR node is required to actively learn the radio space to perform its task intelligently. Similarly, the real learning procedure is enabled by a distinctive and critical process called SS and detection, and it's distinctive to all CR systems. Furthermore, the CR users continuously sense the spectrum and effectively learn the radio space by detecting active incumbent users in the environment.

Spectrum sensing involves the process by which an antenna is used to sense a radio environment [32,8]. The sensed signal is employed to derive a test statistic that will establish if a licensed user is present or not, as illustrated in Fig. 3. spectrum sensing and detection are characterized using statistical detection theories [33,34].

The ED spectrum sensing is the most employed technique due to its simplicity for detecting PUs in a CR environment [32,36–39]. In ED, prior information of the PU is not mandatory to determine whether the channel is busy or not. Besides its simplicity for implementation, it has the benefit of reduced computational complexity [38–41,43]. However, the ED performance suffers the drawback of significant noise uncertainty.

As revealed in Fig. 3, an antenna is used by the CR node for effective radio sensing. Similarly, $r(t)$ is the detected signal and $n(t)$ represents the surrounding noise factor. Taking a statistical approach, to model the radio space, we can define the sensed signal in the context of two hypothetic conditions as expressed in equation (1):

$$r(t) = \begin{cases} n(t); \\ \quad \text{Hypothesis } H_0\text{: if the radio signal is not active} \\ s(t) + n(t); \\ \quad \text{Hypothesis } H_1\text{: if the radio signal is active} \end{cases} \tag{1}$$

where $s(t)$ denotes the sensed radio signal from the antenna as a result of transmissions arising from the environment. The CR node uses the sensed signal $s(t)$ to decide if the active radio user is present or not. Hence a decision is made from the comparative assessment between the specified test statistic ξ and a specified threshold value λ.

In ED, the received signal energy has a component that is calculated within a given time interval T and is used for detecting the test statistic, defined as $T = NT_s$ where T_s is the sampling period of the signal. Similarly, the study in [42], has proved that ED is optimum if $s(t)$ equals zero-mean complex Gaussian.

The test statistic for ED is expressed as,

$$\xi = \int_{t_0}^{t_0+T} r(t)\tilde{r}(t)dt \tag{2}$$

where, $\tilde{r}(t)$ is a complex conjugate acting on $r(t)$ and $t_0 \in \mathbb{R}^+$ denotes the initial arbitrary time. The signal-to-noise ratio (SNR) is denoted as ρ, which is a function of the received signal $s(t)$ when the signal is constantly active during the specified time $t_1 < t \leq t_2$ for t_1, $t_2 \in \mathbb{R}^+$, stated as,

$$\rho = \frac{\alpha^2}{\sigma^2[t_2 - t_1]} \int_{t_0}^{t_0+T} s(t)\tilde{s}(t)dt \tag{3}$$

where α is signal power and σ is the noise power.

For the discrete signal, $r[n] = r(nT_s)$, the test statistic for ED is given by,

$$\xi \approx T_s \sum_{n=0}^{N-1} r[n]\tilde{r}[n] \tag{4}$$

where N is the component that defines the performance of the energy detector and represents the number of complex variables also known as a time-bandwidth product [43].

The decision standard at the CR node is expressed as

$$d = \begin{cases} 0; \\ \quad \xi < \lambda;\ \text{draws an inference decision about hypothesis } H_0 \\ 1; \\ \quad \xi \geq \lambda;\ \text{draws an inference decision about hypothesis } H_1 \end{cases} \tag{5}$$

where d is the decision from the active CR node, as presented in (5) [42].

2.2. Review of related work

Several traditional SS approaches have been implemented, such as energy detection (ED), waveform detection (WD), eigenvalue-based detection (EVD), and cyclostationary feature detection (CFD) [23,44]. The ED approach evaluates the received signal power and compares it with a standard threshold value, then estimates if a PU is active or not. Nevertheless, the ED performance is heavily compromised with noise uncertainty [45]. WD approach, though highly reliable, gives the correlation of a reference waveform and the transmitted signal. The WD approach records higher efficiency [46] but needs precise prior information of the PU. In practice, the CR users are unable to possess prior information of the PU and thus unsuitable for blind signal detection. EVD approach initially proposed by the authors in [4], performs excellently under low SNR circumstances but has the drawback of increased computational complexity [8]. Similarly, the CFD approach [47,48] estimates radio signals in the form of cyclostationary features [49]. Cyclostationary features exhibit random processes with periodic statistics [48]. It enhances channel estimation or synchronization [45]. CFD approaches have the advantages of being utilized for blind detection [50,51].

However, the detection precisions of these techniques are heavily reliant on the apparent noise uncertainty [52,53]. Moreover, those techniques involve complete or partial knowledge of the previous PU's [54]. Rather than requiring features for prior knowledge, machine learning (ML) approaches are currently employed [55] to effectively learn the system communication data. More essentially, ML methods can also learn the underlying patterns of the signals to increase detection precision.

Recently, to solve the SS task, machine learning (ML) approaches have been adopted for SS [56–65]. In [65], the authors implemented a support vector machine (SVM) in a cooperative sensing scheme. The resulting cooperative sensing scheme leads to a more improved sensing efficiency. The adoption of ensemble learning techniques is another method of attaining improved sensing efficiency. In recent times, ensemble classifiers have displayed favorable results in several detection issues [66–71]. The Ensemble classifiers employ several learning procedures to attain a detection efficiency greater than the existing base learners [72–74].

Within the context of reinforcement learning (RL) for SS, in [75], the authors modeled a SU with a Q-learning algorithm to learn the activities of multiple SUs and subsequently select individual users for improved sensing accuracy. However, limited or no consideration was made concerning the sensing ability of SUs while selecting neighboring nodes for cooperation, which result in low sensing accuracy. AN additional RL algorithm is the Multi-armed bandit (MAB) [76], the authors in [77], modified the performance gain of the traditional greedy approach as a reward for the MAB and applied it to compressive sensing. The investigational result shows an improved performance compared to the traditional greedy approach.

Similarly, to permit CRs to actively learn from an environment, several authors have adopted Deep RL processes for spectrum sensing in CR [56–60,62–64,78]. Particularly, in [23] an algorithm using deep Q-learning (DQL) is employed to select sensing channels and eventually access the maximum sensing rate. In [79], also deep Q-network (DQN) is used for spectrum sharing and power control. The authors in [80], proposed a DQN experimented on a reservoir computing framework using a recurrent neural network (RNN) for temporal prediction of the dynamic patterns of the SS. In [81], the authors used a convolution neural network (CNN) based cooperative SS and assessed energy-controlled data protection in CRs. In [82] hybridization of the DQN and Replicator Dynamic (DQN-RD) algorithm was demonstrated, where the RD utilizes the evolutionary game theory principles in processing the reward function of the RL. Their method was used for analyzing the dynamic spectrum access strategy. Another study was also inspired by using CNN for constructing models that can classify the SS in CR networks [83].

3. Deep reinforcement learning

Deep RL combines the observation function of DL with the active decision-making capability of RL. It is a technique of artificial intelligence that mimics human thinking. The basic structure of Deep RL is displayed in Fig. 4. DL obtains the system information by observing the environment and provides the system state information for the current environment. The RL then maps the present state information to the equivalent action and estimates values using the expected accumulated return [84,85].

The mathematical fundamentals and theories of RL are introduced subsequently.

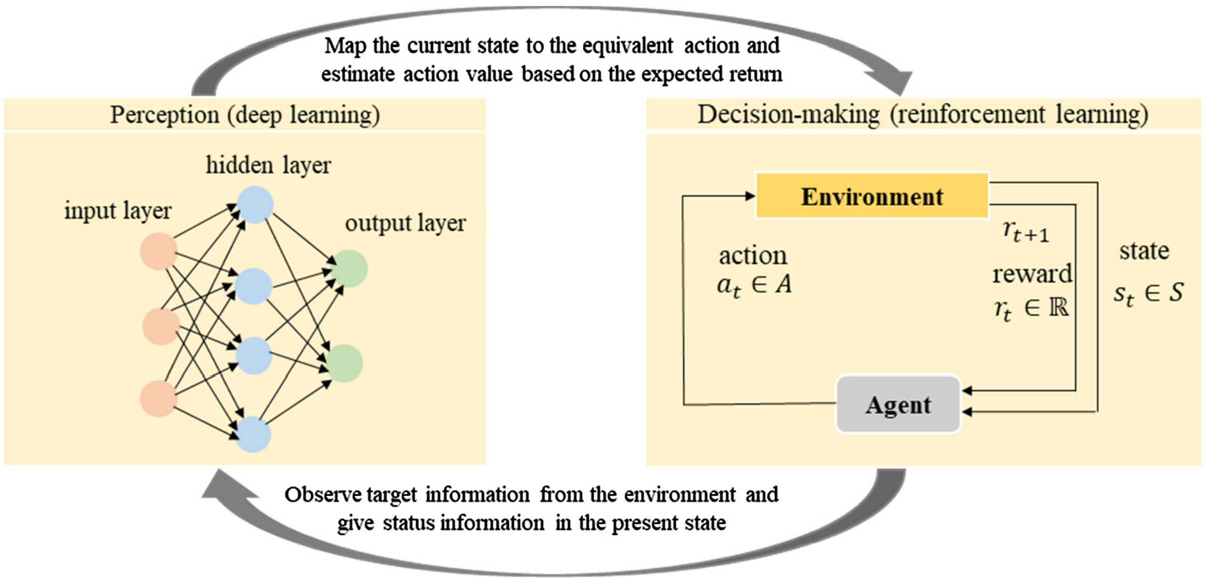
Map the current state to the equivalent action and estimate action value based on the expected return
Perception (deep learning)
hidden layer
input layer
output layer
Decision-making (reinforcement learning)
Environment
action
$a_t \in A$
r_{t+1}
reward
$r_t \in \mathbb{R}$
state
$s_t \in S$
Agent
Observe target information from the environment and give status information in the present state

Fig. 4. Deep RL structure using a multilayer perceptron. Adapted from [86].

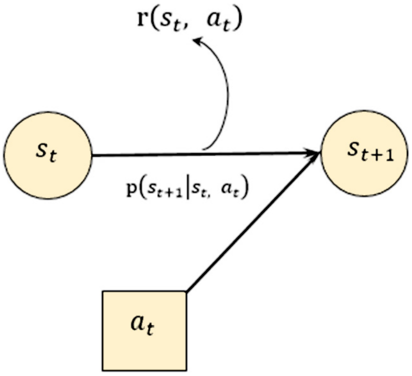

Fig. 5. Markov Decision Process. Reprinted from [87].

3.1. Markov Decision Process (MDP)

The MDP contains Markov characteristics that provide the basic foundation of RL. In a Markov property, the impending future of the structure depends merely on the present state, with the agent not requiring a complete history of the system [87]. Fig. 5 denotes an MDP process, in each time step t, an action a_t is performed on a process in the present state s_t, and there is a transition to the next state s_{t+1}. A Reward r_t is acquired in this transition.

An MDP can be defined as:

$$P(s_{t+1} \mid s_0, a_0, \cdots, s_t, a_t) = P(s_{t+1} \mid s_t, a_t) \tag{6}$$

where P denotes the state transition probability. Thus, from the current state s, for every time step t, whereby the agent performs an action a_t and receives a pair of reward r_t at state s_t, and the next state s_{t+1} is updated appropriately. Consequently, to process the received reward value, a value function and optimum policy are required.

- *Value function and optimum policy*: To effectively maximize the future accumulative reward after the present state during any time horizon T, the reward R_t according to [88] is given by:

$$R_t = r_{t+1} + \gamma r_{t+1} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{7}$$

where $\gamma \in [0, 1]$ accounts for the discount factor and can only take a value of 1 in very rare MDPs condition. To determine an optimum policy, some RL algorithms are established on the value function $V(s)$, which signifies how useful it is for an agent to attain a specified state s. Such a given function depends mainly on the real policy π [89]:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, \pi\right], \tag{8}$$

$$V^*(s) = \max_{\pi \in \Pi} V^{\pi}(s). \tag{9}$$

The variable Q represents the Q-value function (action-value) which is the value of performing an action a in a given state s within a policy π [89], given as

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a, \pi\right]. \tag{10}$$

For a Q-learning algorithmic process [89], the active Q function can be defined in the context of Bellman's equation:

$$\begin{aligned} Q^{\pi}(s, a) = \sum_{s_{t+1} \in S} & T(s, a, s_{t+1}) \big(R(s, a, s_{t+1}) \\ & + \gamma Q^{\pi}(s_{t+1}, a = \pi(s_{t+1}))\big), \end{aligned} \tag{11}$$

where T, is the time horizon, γ represents discount factor, s is the present state, s_{t+1} is the next state after the agent performs an action a in state s. R is the reward function.

The optimal Q-value is given by

$$Q^*(s, a) = \max_{\pi \in \Pi} Q^{\pi}(s, a). \tag{12}$$

The optimal policy π^* specify the largest accumulative reward in the long-term [89]:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s). \tag{13}$$

The function Q^* denotes the Q-value function, whose dependent variables are; state s and action a. The basic principles of RL are presented in the following sub-section.

3.2. Reinforcement learning

RL is employed to evaluate a behavioral pattern and a policy that maximizes an acceptable criterion. Nevertheless, the learning process requires a huge amount of data samples to attain the optimal policy. A policy can mathematically be denoted as π and can be defined as the mapping from the state of an environment to a specific action of an agent. Furthermore, long-term accumulation of rewards is acquired using the interaction between the agent and the specified environment, using a trial and error strategy. To implement these tasks, an RL framework consists of an active decision-maker, known as the agent, functioning in an environment that is modeled by a state s_t. Likewise, the agent is capable of taking a certain action a_t, which depends on the present state s_t. When an action is chosen at time t, the agent then gets a scalar reward r_{t+1} and transition to the next state s_{t+1} as a result of the present state and the selected action, as depicted in Fig. 4. Most RL approaches are related to dynamic programming algorithms, Monte Carlo techniques, and temporal difference (TD) learning, as explained in the following subsections [87].

3.2.1. Dynamic programming

Essentially, RL could be described in terms of dynamic programming when the system dynamics are unknown beforehand. RL techniques may be classified as model-free or model-based (i.e., if they develop a model representing the transition probability and the reward functions $p(s_{t+1} \mid s, a)$ and $r(s, a)$ of the fundamental MDP). Consequently, dynamic programming approaches may be considered as a specialized and perfect case of indirect or model-based RL techniques. If a system dynamics is unknown, the model-based approaches can implement the system online. In a discrete situation, this can be achieved using a simple accumulative method established on the principle of maximum likelihood. In contrast, direct or model-free approaches do not develop models. The hidden transition and reward parameters $p(s_{t+1} \mid s, a)$ and $r(s, a)$ are usually not evaluated, instead, a value parameter V is locally updated during the experiment. This method has the advantage of memory utilization and, traditionally RL was limited only to direct approaches. The main idea behind direct RL involves the local optimization of a policy after every action with an environment. Since it consists of a local update, a global estimate of a policy is not necessary. In reality, most direct RL methods cannot operate on the policy directly, instead, they give an approximation of the value function. In general, dynamic programming employs value functions for optimizing good policies [90]. The optimal policy can simply be obtained from the value function, v_* or quality function q_*, based on the Bellman's optimality equations [90]:

$$v_*(s) = \max_a \mathbb{E}\big[R_{t+1} + \gamma v_* \,(S_{t+1}) \mid S_t = s, A_t = a\big] \tag{14}$$

$$v_*(s) = \max_a \sum_{S_{t+1},r} p(\,S_{t+1}, r \mid s, a)\big[r + \gamma v_*(S_{t+1})\big] \tag{15}$$

or

$$q_*(s,a) = \mathbb{E}\big[R_{t+1} + \gamma \max_{a_{t+1}} q_* \,(S_{t+1}, a_{t+1}) \mid S_t = s, A_t = a\big] \tag{16}$$

$$q_*(s,a) = \sum_{S_{t+1},r} p(\,S_{t+1}, r \mid s, a)\big[r + \gamma \max_{a_{t+1}} q_*(S_{t+1} a_{t+1})\big]. \tag{17}$$

The above Bellman equation is often explored for deriving the dynamic programming algorithms, this implies that the RL values and functions are converted into update procedures to obtain optimized value functions [90].

3.2.2. Monte Carlo techniques

These techniques involve executing several trajectories of all possible states s, and approximating $V(s)$ as the mean accumulative rewards from all the trajectories. During every trial, the transitions and rewards are recorded by each agent and then produce an estimated value of the states based on a discounted reward structure. The specific value of every single state converges to $V^{\pi}(s)$ due to a policy π.

Therefore the key feature behind Monte Carlo techniques is the continuous estimates of a state value based on an accumulated reward resulting from a series of trajectories.

Additionally, let $(s_0, s_1, \ldots, s_N)$ represent a trajectory in line with a policy π and an unidentified transition variable $p()$ and also let $(r_1, r_2, \ldots, r_N)$ represent the rewards from the trajectory. Employing the Monte Carlo technique, the updates of N parameters, $V(s_t), t = 0, \ldots, N-1$, are simplified as

$$V(s_t) \leftarrow V(s_t) + \alpha(s_t)\big(r_{t+1} + r_{t+2} + \cdots + r_N - V(s_t)\big) \tag{18}$$

where $\alpha(s_t)$, represents the learning rates that gradually converge to 0 for a predefined during the iterations. This implies that V would converge to V^{π} using established assumptions [91].

3.2.3. Temporal Difference (TD) approaches

The update scheme in the equation (6) earlier discussed the Monte Carlo technique, which can be modified temporal difference (TD) error, resulting in a better incremental process [87]: The TD approach can be defined as:

$$V(s_t) \leftarrow V(s_t) + \alpha(s_t)(\delta_t + \delta_{t+1} + \cdots + \delta_{N-1}) \tag{19}$$

where TD error δ_t can be defined by

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t), \quad t = 0, \ldots, N-1. \tag{20}$$

The error δ_t could be deduced as a factor differentiating the present approximation $V(s_t)$ and the corrected approximation $r_{t+1} + V(s_{t+1})$. It is evaluated immediately after the transition (s_t, r_{t+1}, s_{t+1}), resulting in an online update learning scheme (7). Hence V is updated before the trajectory ends [92].

$$V(s_\iota) \leftarrow V(s_\iota) + \alpha(s_\iota)\delta_t \quad \iota = 0, \ldots, t. \tag{21}$$

The basic TD algorithm is denoted as TD(0).

The algorithm depends on the relationship between the truly received reward and the expected reward from the earlier approximations. If the approximated values of $V(s_t)$ and $V(s_{t+1})$ are precise in the current state s_t and next state s_{t+1}, then we can have the following [87]:

$$V(s_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots, \tag{22}$$

$$V(s_{t+1}) = r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} + \ldots. \tag{23}$$

Hence, the compact representation of equations (10)–(11) are expressed as follows,

$$V(s_t) = r_{t+1} + \gamma V(s_{t+1}) \tag{24}$$

$$V(s_t) \leftarrow V(s_t) + \alpha\big[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\big] = V(s_t) + \alpha\delta_t. \tag{25}$$

The updated equation instantly reveals the relationship between TD approaches, Monte Carlo techniques, and dynamic programming [87].

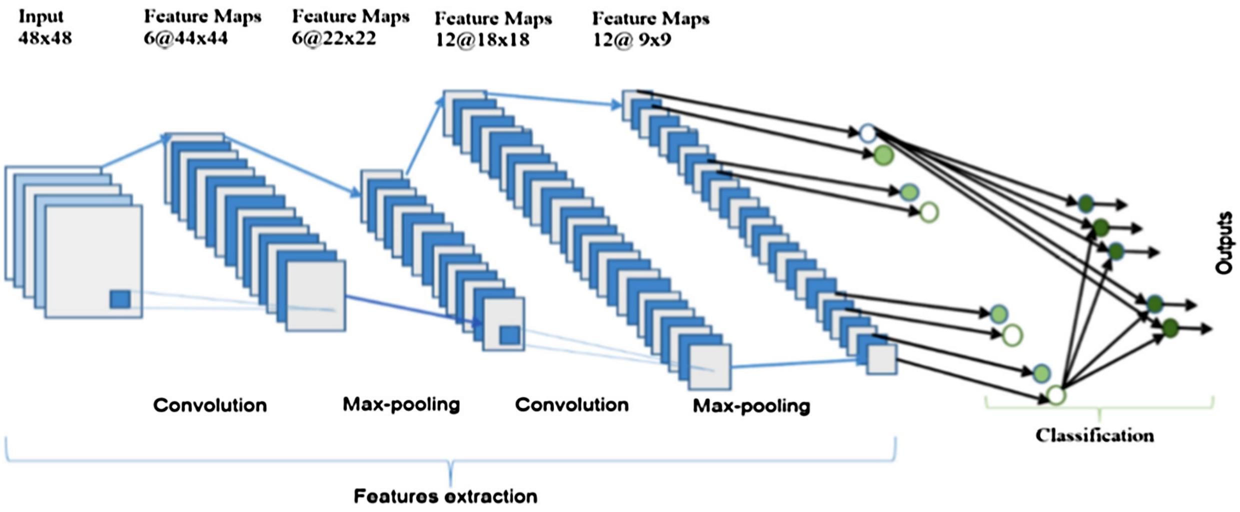

Fig. 6. System architecture of CNN [98].

3.2.4. POMDPs

Even though the majority of the real-life scenarios are represented as an MDP, in practice, most agents are unable to completely learn the environment. Often they lack sufficient knowledge of the environment. POMDPs are designed specifically to address this type of condition where the agent possesses only partial information of the system to control.

A POMDP is defined as an MDP in which the agent lacks knowledge of the actual environment: it may possess only a partial view of its state [93]. A POMDP is expressed as a tuple that is dependent on these variables $(S, A, \Omega, T, p, O, r, b_0)$ where: S represent the state; A is the action; Ω denote the observation; T signify time; p represent transitional probabilities; O represent observation probabilities; r denote the functional reward and b_0 is the initial probability distribution through the states.

3.3. Deep learning

DL [22] is a subfield of Artificial Intelligence that involves the interconnection of several layers of neural networks for learning data representation while performing a specific or multi-modal task. In the context of the current study, the deep learning algorithm in the perspective of deep RL is often found in the policy and learning algorithm (actor and critic components of the agent). DL aims to curb traditional feature engineering by automatically learning from a given data. It is important to note that DL is Artificial Neural Network (ANN) structure originally inspired from biological neurons. Different learning methods employ several forms of DL architectures. Examples of such architecture include; Deep Neural Networks (DNN), CNN, and Recurrent Neural Network (RNN) architectures comprise of Long Short Term Memory (LSTM), Gated Recurrent Units (GRU), amongst others.

3.3.1. Deep neural networks

Artificial neurons are the basic constituent for developing ANNs, which attempt to imitate human intelligence. The main computational component (neuron) is referred to as a node that gets input signals from exterior sources and possesses inner components (containing weights and biases used for learning in the training process) to create outputs. The fundamentals of ANN were discussed in [94,95]. A typical ANN is an interconnection of three network layers: the input, hidden, and output layers. An ANN can exist in two broad forms: feed-forward neural networks (FNN) and RNN. An FNN is a branch of ANN that does not have a feedback loop in the learning process and can be described into categories; multi-layer perceptron (MLP) and convolutional neural networks (CNN). An RNN is the direct opposite of the FNN, as it requires a feedback loop in the neural network architectural setup. Hence, DNN relies on several amounts of neural network layers within the hidden layer, which can exist as either FNN or RNN. For an in-depth study about MLP, we refer the reader to [94–96]. One important element used in training DNNs is the learning rate, which is the step size used in training to increase the training speed. Nevertheless, choosing a value for the learning rate becomes delicate. For instance, a large value of η makes the system diverge rather than converge. Similarly, a small value of η will reduce the speed of convergence. Furthermore, it can lead to the problem of local minima. A typical remedy for this issue is to decrease the learning rate for training [97]. We refer the reader to [98] for more details on DNN.

3.3.2. Convolutional neural networks

CNN is a popular example of ANN, with an extensive series of applications particularly in speech recognition and computer vision. It comprises of artificial neurons that are structured into three interrelated layers: the input, hidden, and output layers. The unique characteristic of CNN is that the input (image, signals, structured data) and kernel weights are convolved based on an element-wise multiplication and consequently summed up to obtain an effective feature map. Each neuron has active weighted inputs, which defines an output with a specified input, and one definite output. Compared to DNNs, CNNs possess numerous benefits, in addition to being extremely optimized for 2D and 3D image processing, they are very effective for learning and extraction of 2D structures [98]. The network layers have a max-pooling that is suitable for absorbing shape dimensions. CNNs are less prone to the problem of diminishing gradient since, in practice, this problem can be minimized using normalized initialization [99] and in recent times, through Batch Normalization [100]. Similarly, they can generate extremely optimized weights since the gradient learning procedure trains the entire network to directly reduce the error through back-propagation. The complete system architecture is shown in Fig. 6 consisting of two basic parts: the feature extraction and the classification. The CNN structure is made up of three layers, such as convolution, max-pooling, and classification. A defining characteristic of CNNs is the operation of convolution. The convolutional layer is the first layer that extracts features from the images, and the other convolutional layers process feature maps from the previous layer to obtain more informative object representation. A CNN typically uses several small-size kernels $n \times n$

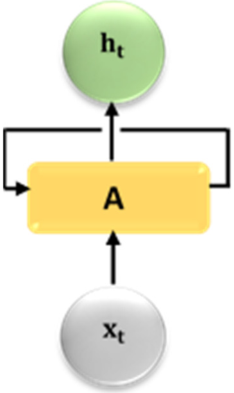

Fig. 7. Basic structure of RNN using a loop. Adapted from [98].

whose input weights and bias $\{\boldsymbol{W}_{\boldsymbol{i}}, b_i\}$ for $i = 1, 2, \ldots, J$ are often used for generating multiple feature maps after extracting distinctive features of the input. Pooling layers are typically used after convolutional layers, to further decrease the complexity in addition to improving the network robustness using down-sampling principles (sum-pooling, average pooling, and max-pooling). A fully connected network can be referred to as an MLP or dense layer, where every output neuron from a pooling layer is connected to every one of the output neurons. We refer the reader to [98] for comprehensive details on CNN.

3.3.3. Recurrent neural networks

An RNN is an interconnection of neural network nodes with a feedback loop containing a gated mechanism (internal memory). RNN methods can cope with steady-size input vectors (for example, a video, image frame, or temporal data), which produces a sequence or temporal prediction. Moreover, RNN can function with only a fixed amount of computational layers (i.e., a fixed number of layers in the network). The illustration of an RNN is shown in Fig. 7.

Jordan and Elman have proposed different types of RNN [101, 102]. Mathematically stated as Elman model [101]:

$$h_t = \sigma_y(w_h x_t + u_h h_{t-1} + b_h), \tag{26}$$

$$y_t = \sigma_y(w_y h_t + b_h). \tag{27}$$

Jordan model [102]:

$$h_t = \sigma_y(w_h x_t + u_h y_{t-1} + b_h), \tag{28}$$

$$y_t = \sigma_y(w_y h_t + b_h), \tag{29}$$

where x denotes the input vector, h represents the vectors for the hidden layer, y are output vectors, while w and u represent weighted matrices, b denotes the bias vector. For a detailed study on RNN, we refer the reader to [98]. The examples of two famous RNN architectures described in the subsections below.

3.3.3.1. Long short-term memory LSTM is a variety of RNN, that was initially introduced by the authors [103] that are capable of learning long-term dependencies, principally for sequence prediction challenges. As stated by the study in [104], the LSTM network is structured like a chain configuration integrating cell states and gate mechanisms (input gate, forget gate, and output gate). The basic structure of an LSTM is shown in Fig. 8. We refer the reader to [105] for in-depth details on LSTM and its applications in the following areas: language modeling, machine translation, image captioning, question answering, video-to-text conversion, amongst others.

3.3.3.2. Gated recurrent units GRUs is a variant of RNN (LSTM) but has a slight variation [106] to its gating mechanism. A GRU consists of two gates: reset gate and update gate. GRU has gained popularity in the research community of the RNNs. The main motivation behind this attractiveness is due to its simplicity and low computational cost. GRUs represents simpler forms of RNN techniques compared with LSTMs in the context of the computational cost, topology, and system complexity [106]. A detailed study of the GRU is reported in the paper [98].

3.4. Deep reinforcement learning algorithms

Deep RL problems can be expressed as optimization, scheduling, supervision, and control problems [107]. The solution techniques can be model-based (or model-free) and value-based (policy-based), as illustrated in Fig. 9. In RL, a model can be defined as the dynamic states of an environment and how these states lead to a reward. This involves knowing the MDP to create a model representing the probability of the state transition (i.e. the probability of moving from the current state to the next state after an action). A model-free approach is an algorithm that does not use the transition probability distribution and the reward function associated with the MDP. This implies that in a model-free scenario, we do not need to know the inner working of the system. However, in a model-based approach, the reward function must be defined and computed to obtain optimal actions using the model directly.

Model-based Deep RL is powerfully influenced by modern control theory which is often explained within the context of different disciplines. Unlike model-based, model-free Deep RL disregards the model and has less concern about the internal mechanisms. Model-based Deep RL has the benefit of being efficient and simple. For instance, if it becomes suitable to approximate a given space as linear, model-based will take considerably fewer samples in learning the model. Nevertheless, model-based techniques are more difficult than model-free techniques. If sampling may be completed using a computer-based simulation, model-free techniques finish faster. Similarly, to simplify computational complexity, model-based techniques have more approximations and assumptions and hence, may constraint themselves to particular kinds of tasks. The value-based methods are built on value and policy functions.

Therefore, we emphasize one famous value-based method, the Q-learning algorithm [108]. Similarly, the basic features of the deep Q-network (DQN) algorithm [109] are considered, which was able to attain a superhuman level of control with the ATARI game. Furthermore, a complete overview of the typical Deep RL algorithms presented in Fig. 9, is not within the scope of this review. For in-depth details on the typical Deep RL algorithms, we refer the reader to [89,110]. To keep the review as concise as possible, we emphasize only one policy-gradient method (actor-critic) and one popular model-based method (Probabilistic Inference for Learning Control (PILCO)).

3.4.1. Q-learning algorithm

It employs the Bellman equation for learning and computing the optimum Q-value function [89], whose distinctive solution is given as $\mathcal{Q}(s,a)$:

$$\begin{aligned} Q^{\pi}(s,a) = \sum_{s_{t+1}\in S} & T(s,a,s_{t+1})\Big(R(s,a,s_{t+1}) \\ & + \gamma Q^{\pi}\big(s_{t+1}, a = \pi(s_{t+1})\big)\Big). \end{aligned} \tag{30}$$

For Q-learning, the maximum action-value is considered

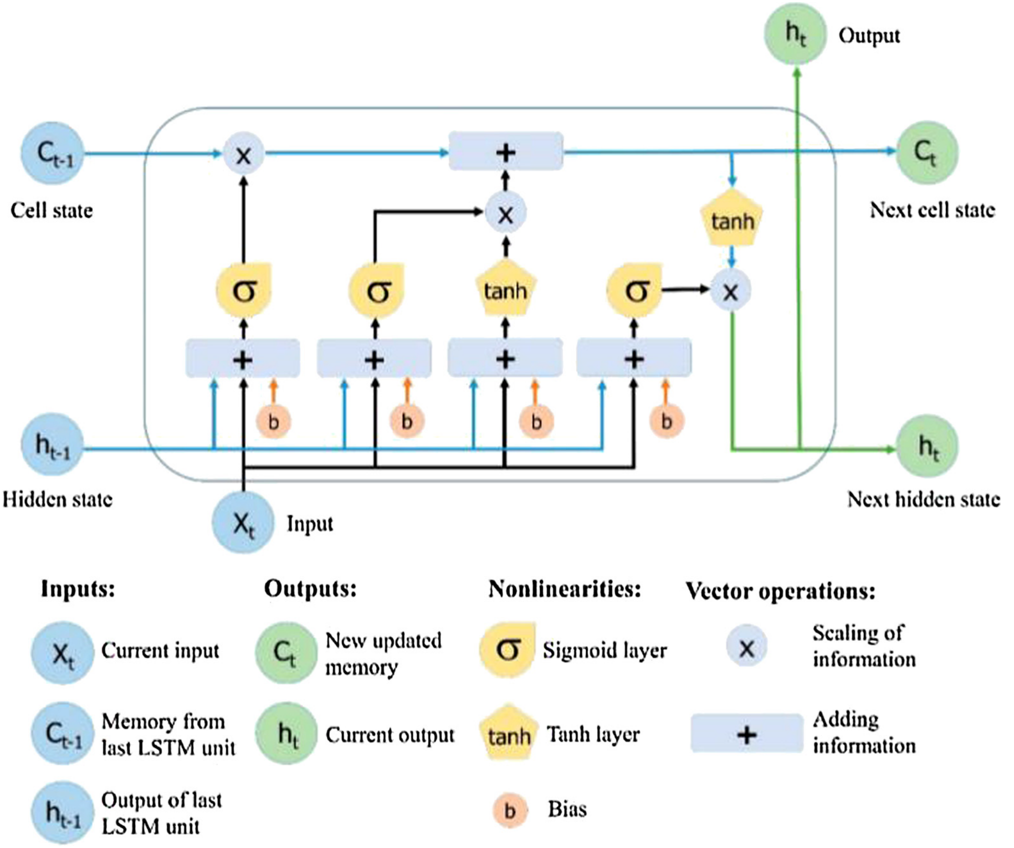

Fig. 8. Structure of LSTM. Reprinted from [105].

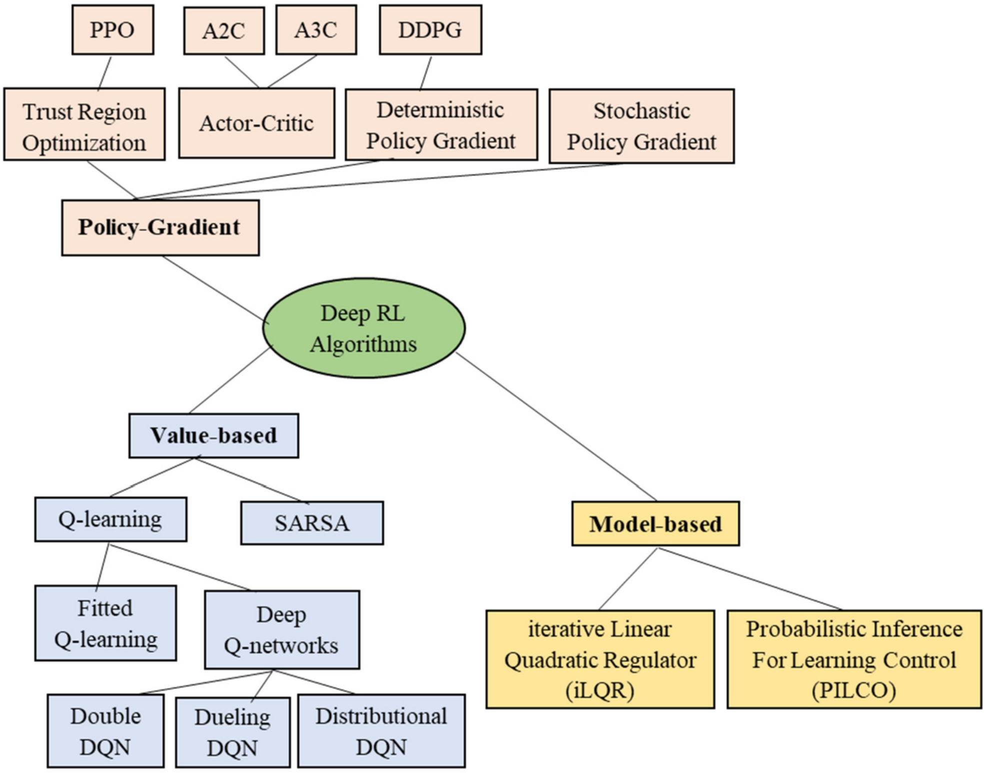
PPO
A2C
A3C
DDPG
Trust Region Optimization
Actor-Critic
Deterministic Policy Gradient
Stochastic Policy Gradient
Policy-Gradient
Deep RL Algorithms
Value-based
Q-learning
SARSA
Model-based
Fitted Q-learning
Deep Q-networks
iterative Linear Quadratic Regulator (iLQR)
Probabilistic Inference For Learning Control (PILCO)
Double DQN
Dueling DQN
Distributional DQN

Fig. 9. Typical Deep RL algorithms [107].

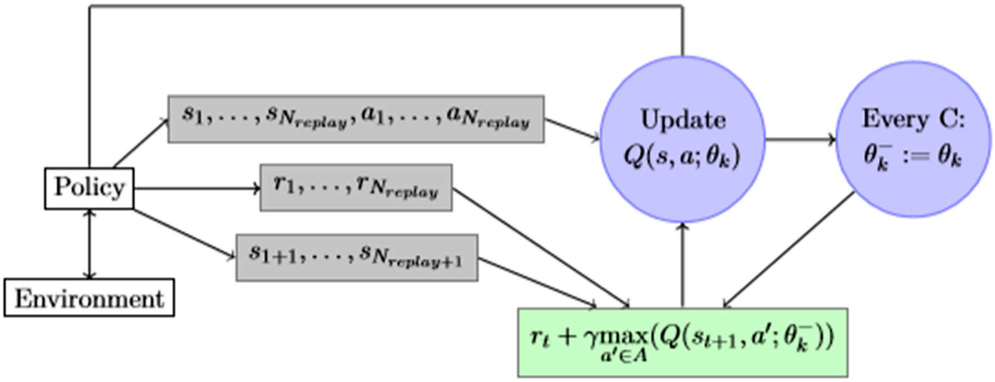
$s_1, \ldots, s_{N_{replay}}, a_1, \ldots, a_{N_{replay}}$
Policy
$r_1, \ldots, r_{N_{replay}}$
$s_1+1, \ldots, s_{N_{replay}+1}$
Environment
Update $Q(s, a; \theta_k)$
Every C: $\theta_k^- := \theta_k$
$r_t + \gamma \max_{a' \in A}(Q(s_{t+1}, a'; \theta_k^-))$

Fig. 10. A sketch depicting the DQN algorithm [89].

3.4.2. *DQN algorithm*

This algorithm introduced by [109] can perform excellently in online games. It utilizes two basic heuristics for limiting uncertainties:

Firstly, the central Q-network function is given by,

$$Y_k^Q = r + \gamma \max_{a_{t+1} \in A} Q\left(s_{t+1} a_{t+1}; \overline{\theta}_k\right), \tag{31}$$

where $\overline{\theta}_k$ represents the limiting factors, r is the reward, γ is the discount factor, s_{t+1} is the next state and a_{t+1} is the next action.

Secondly, in an online context, the DQN uses a replay memory [111] which stores all previous time steps data. This technique permits updates covering a wider range of its state-action space. A sketch describing the DQN algorithm is shown in Fig. 10.

3.4.3. *Actor-critic methods*

The actor-critic design comprises two sections: the actor and the critic [112]. It is a common method where an actor updates a policy network with policy gradients and the critic performs a value function estimate for the present policy [112].

Several approaches combine both on-policy with off-policy figures for policy estimation [113]. The *Retrace*(λ) algorithm [114] proves more reliability due to the optimum usage of samples accumulated from close on-policy behavioral policies. The same method was adopted in the actor-critic systems described by [115,116]. These systems are sample-efficient and the advantage of using the replay memory, with computational efficiency as they employ multi-step returns that increases learning stability and improves the speed of backward propagation of the reward in time.

The off-policy gradient for the stochastic case in the policy improvement phase is:

$$\nabla_w V^{\pi w}(s_0) = \mathbb{E}_{s \sim \rho^{\pi_\beta}, a \sim \pi_\beta}\left[\nabla_\theta\left(\log \pi_w(s, a)\right) Q^{\pi_w}(s, a)\right], \tag{32}$$

where β the behavior policy is usually distinct from π, and the critic, having parameter θ, approximates the expected value function $Q(s, a; \theta)$ for the present policy π. In practice, this method usually behaves accurately but the analysis of its convergence becomes complex due to the use of a bias-based policy gradient estimator. In this case, an improved method to perform the on-policy approach without using experience replay is to employ asynchronous techniques (such as Asynchronous Advantage Actor-critic A3C), where several agents are trained and implemented through parallel asynchronous processing [117]. In reality, many policy gradients' approaches effectively employ undiscounted state patterns, without affecting their performance [118].

3.4.4. *Probabilistic inference for learning control*

For example, in PILCO, the authors in paper [119] exploit Gaussian processes for learning probability modeling of the system dynamics. This can then apply the uncertainty explicitly for policy evaluation and planning to achieve an excellent sample efficiency. Nevertheless, the Gaussian processes cannot reliably scale to high-dimensional glitches. One method in scaling planning to high-dimension is to leverage the generalization abilities of DL. For example, the authors in [120] exploit a DL structure through a latent state-space model. Also, the author in [121] adopted Model-predictive control. Another method is using the trajectory optimizer as an instructor instead of a demonstrator: directed policy search [122] performs a few series of actions recommended by a different controller.

Motivated by the successes of RL in issues relating to dynamic control, for instance, the popular Atari game [109], and AlphaGo [123], interested practitioners have shown increased attention to RL-based solutions for wireless communications. For DSA in CR networks, the system model is usually formulated as an MDP [124,125] or a POMDP [126], which depends on whether the system is fully observable or partially observable to the users. In [127], Q-learning is applied to SS order selection, under imperfect sensing, similarly, the authors in [126,128,129] have applied DQN, a typical RL algorithm for several applications such as improving the channel selection accuracy, maximizing the system utility, or minimizing the network blocking probability. Furthermore, to resolve the DSA challenge in decentralized networks, several multi-agent RL schemes were considered in [58,128,130].

4. Deep RL for spectrum sensing in cognitive radio networks

As mentioned in the preceding section, inspired by the successes recorded by RL algorithms, the CR system is perceived as an agent within the context of Deep RL, and using the feedback signal, the system environment could be identified. The optimum decision could be learned, and Deep RL algorithms can be employed for designing an optimum policy for an agent. The spectrum sensing problem could be effectively expressed as an MDP.

4.1. *System model*

A hypothetical cooperative spectrum sensing model that is formulated as Multi-agent MDP called Dec-POMDP is proposed as shown in Fig. 11. Also, any of the Deep RL algorithms discussed earlier can be used to formulate the optimal solution.

The study considers only a single active PU and a CR network consisting of N multiple CR users that observes the action of the active PU. Considering many PUs will lead to sensing complexity in addition to other procedures such as spectrum-handoff and scheduling. The CR nodes sense the spectrum and transmit the resulting outcomes to a fusion center (FC) as shown in the system model (Fig. 11).

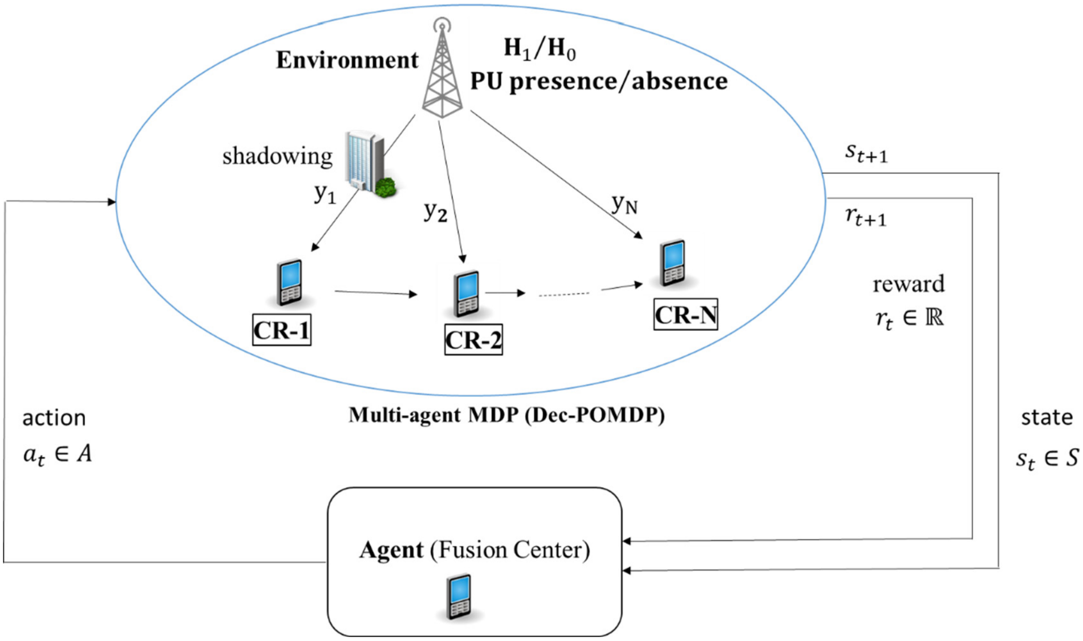

Fig. 11. System Model representation of an RL consisting of an environment (multi-agent MDP) and an agent (Fusion Center).

The proposed system model is presumed to function in a time-slotted fashion. At the start of every time slot, one of the agents, which is also the FC, initiates Deep RL-based cooperative spectrum sensing and combines the local decisions. Using a dedicated reporting channel, the N accounts for the number of CR users as illustrated in the system model. CR users (agents) constantly sense the spectrum and inform the FC about their local decisions [4]. Typical spectrum sensing issues such as shadowing and multiple path fading could be minimized by employing spatial diversity through cooperative SS [131]. In this circumstance, the CR user broadcasts a request for cooperative SS to all the neighbors. The agent then combines all the individual local decisions and gives the final binary decision of the PU's presence in the environment.

Each active CR user utilizes an energy detection pattern for spectrum sensing due to its simplicity. Before SS the CR nodes or SUs are required to have established the portion of the radio spectrum to concentrate their sensing potentials. This task can be realized through an effective sensing policy. Similarly, the occupancy model for the PU in addition to the time slot procedure for sensing constitutes a major role in improving the spectrum sensing policy.

4.1.1. PU activity

A model representing the activity of the PU is paramount for designing a sensing policy, with the purpose of defining precisely the spectrum occupancy of the PU over time. Typically the activity of the PU is adopted to be time slotted so that the active PU is either randomly idle or active for the total period of a given time slot.

For the effective transitions between idle and active PU states, the most typical stochastic models are the widely held time-independent model [132–134], and the classical Gilbert-Elliot model [135–144]. With the time-independent model, the active state of the radio spectrum changes independently through time slots while in the classical Gilbert-Elliot model an assumption is made for the spectrum to change according to a Markov model, as displayed in Fig. 12. The PU either moves into a new state or is kept in a constant state. The transition probabilities are specified on the edges.

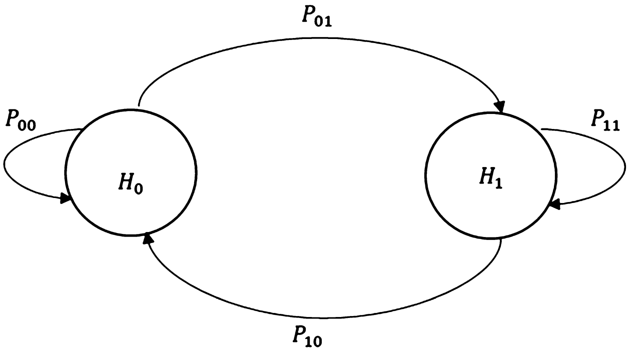

Fig. 12. Markov model containing two states for the PUs activity.

4.1.2. SU activity

The process that the receiver at the sensing node adopts to select the decision rule is based on signal detection theory [145]. The SS problem could be defined by way of a source releasing two likely outputs at several time instances. The corresponding outputs are called hypotheses. H_0 is said to be a null hypothesis if it signifies a zero (which depicts PU not active) but H_1 is the opposite hypothesis if it signifies a one (PU active).

Let [136] represent the not active and active PU states respectively. The sensing decision relies on the detection probability (P_d) and the false alarm probability (P_{fa}). The conditional probabilities for the sensing decisions using the active PU state and CR actions are specified as [145]:

1. $P(decide\ H_0 \mid H_0\ true) = 1 - P_{fa}$ (33)
2. $P(decide\ H_0 \mid H_1\ true) = 1 - P_d$ (34)
3. $P(decide\ H_1 \mid H_0\ true) = P_{fa}$ (35)
4. $P(decide\ H_1 \mid H_1\ true) = P_d$ (36)

Notice that for conditions 1 and 4, the receiver at the sensing node makes an accurate decision, while for conditions 2 and 3, the receiver at the sensing node makes an error. Condition 2 indicates a miss-detection, while condition 3 refers to a false alarm, and condition 4 is called a detection probability.

The state of a channel can be defined in the context of the time slot horizon as described below;

$$x_it = \begin{cases} 0 & \text{if the channel is in 0 state in time slot } t \\ 1 & \text{if the channel is in 1 state in time slot } t \end{cases} \tag{37}$$

4.2. Multi-agent MDP (Dec-POMDP)

In most multi-agent RL problems, the agents (SUs) are not able to observe the full state of the environment. We considered the POMP structure since the model has been designed specifically to resolve situations where the agent can only have partial information on the environment. As a result, the POMP structure perfectly matches the problem of SS in CR networks. For our system model, we adopted a cooperative multi-agent MDP known as Dec-POMP, which is an extension of POMP.

4.3. Channel sensing strategy

An assumption is also made that the switching pattern of the channel is not known to the CR users. To transmit each data successfully, each CR node would have to obtain the switching pattern of the channel from its observation of the channel. For our proposed system model, the states of the channel are assumed to be switching dynamically within the range [0, 1]. The switching strategy of the channel states is based on the concept highlighted by Ahmad and co-workers [136], which can be varied at each time slot and the state remains constant over a specified time slot.

For our Dec-POMD problem formulation, each CR user represents the agent. The agent in the state s_t performs an action a_t i.e., sensing a channel or frequency band, takes a local decision d based on a received test statistics ξ, and gets a reward $R(s, a, s_{t+1})$. The reward is the channel identified as vacant by the CR user in time slot t. This leads to a new state s_{t+1}. The fusion center (which is also an agent that initiates the Deep RL-based cooperative SS) gives a binary decision by fusing the results of the test statistics obtained from the cooperative CR users for the channel. Normal fusion rules like the likelihood ratio test may be used. In a scenario whereby a channel is sensed in an idle state ($x = 0$), the CR transmits and receives a scalar reward, which translates to feedback from the environment, and a function of the channel condition. Otherwise, the CR would not transmit, receives no reward, and then wait until $t + 1$ to execute a new action on the environment.

A Dec-POMDP for n CRs is defined with a tuple ($S, A, P, \ \Omega, O, R$) [87], where:

S represent the system states defining the likely configurations of every CR user;
$A = (A_1, \dots, A_n)$, is the set of all joint actions performed by CR$_i$;
$P = S \times A \times S \to [0, 1]$, is the transition probability function; $P(s, a, s_{t+1})$ for all CRs;
$\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_n$, is the CR's set of observations;
$O = S \times A \times S \times \Omega \to [0, 1]$, is the observation of all CRs;
R is the functional reward for all CRs expressed as $R(s, a, s_{t+1})$.

Each CR$_i$ maximize its total estimated reward defined as

$$R_i = \sum_{t=0}^{T} \gamma^t r_i^t \tag{38}$$

where γ is the discount factor while T denotes the time horizon.

A specific policy π_i for a CR$_i$ in a Dec-POMDP is formerly defined as the mapping from a CR's previous local observations $\overline{o}_i = o_i^1, \dots, o_i^t$ to an action $a_i \in A_i$. As the optimum policy is relatively probabilistic, each CR's must consider the main history of individual observation and the history of its actions, However, in reality, each agent can only partially observe the environment, a specific policy π_i for individual CR$_i$ in a Dec-POMDP could similarly be expressed as the representation of a CRs' belief states to the action space. Thus, we model our system based on a belief state.

A belief state defines the probability distribution of the state space [87,146]. As previously stated, an adequate information state should be updated when an action is executed. Hence, b represent the belief state of a POMDP process due to an action a and observation o, which is expressed as b_o^a, where:

$$b_o^a(s_{t+1}) = \Pr(s_{t+1} \mid b, a, o) \tag{39}$$

$$= \frac{\Pr(s_{t+1}, b, a, o)}{\Pr(b, a, o)} \tag{40}$$

$$= \frac{\Pr(o \mid s_{t+1}, b, a,) \Pr(s_{t+1} \mid b, a) \Pr(b, a)}{\Pr(o \mid b, a) \Pr(b, a)} \tag{41}$$

$$= \frac{\Pr(o \mid s_{t+1}, b, a,) \Pr(s_{t+1} \mid b, a,)}{\sum_{s_{t+1} \in S} \Pr(s_{t+1} \mid b, a) \Pr(o \mid s_{t+1}, a)} \tag{42}$$

$$= \frac{O(o \mid s_{t+1}) \sum_{s \in S} \Pr(s_{t+1} \mid a, s) \Pr(s)}{\sum_{s \in S} \sum_{s_{t+1} \in S} O(o \mid s_{t+1}) \Pr(s_{t+1} \mid a, s) \Pr(s)} \tag{43}$$

$$= \frac{O(o \mid s_{t+1}) \sum_{s \in S} p(s_{t+1} \mid s, a) b(s)}{\sum_{s \in S} \sum_{s_{t+1} \in S} O(o \mid s_{t+1}) p(s_{t+1} \mid s, a) b(s)}. \tag{44}$$

Which leads to

$$b_o^a(s_{t+1}) = \frac{O(o \mid s_{t+1}) \sum_{s \in S} p(s_{t+1} \mid s, a) b(s)}{\sum_{s \in S} \sum_{s_{t+1} \in S} O(o \mid s_{t+1}) p(s_{t+1} \mid s, a) b(s)}. \tag{45}$$

The conditional probability can be defined as

$$\omega(b, a, o) = \Pr(o \mid b, a) \tag{46}$$

$$= \sum_{s \in S} \sum_{s_{t+1} \in S} O(o \mid s_{t+1}, a) p(s_{t+1} \mid s, a) b(s) \tag{47}$$

The transition probability for the belief state is defined as

$$\Pr(b_{t+1} \mid b, a) = \sum_{o \in \Omega} \omega(b, a, o) \delta\big(b_{t+1}, b_o^a\big). \tag{48}$$

The reward function can be defined as

$$\rho(b, a) = \sum_{s \in S} r(s, a) b(s). \tag{49}$$

The optimal policy π for the n-CR in a Dec-POMDP is given by:

$$V^{\pi}(s_0) = E\left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t, s_{t+1}) \,|s_0, \pi\right]. \tag{50}$$

Within the context of our proposed system model, that is, Multi-agent MDP (Dec-POMP), the reward is defined according to the reward function formulation in Equation (49).

5. Performance comparison of conventional and artificial intelligence-based spectrum sensing techniques in CR networks

We compared the performance of both conventional and artificial intelligence-learning-based spectrum sensing techniques in CR networks as summarized in Table 1. Conventional spectrum sensing can be divided into four forms; energy detection, cyclostationary feature-based detection, matched filter, and covariance-based spectrum sensing.

Table 1
Performance comparison of conventional and artificial intelligence-based spectrum sensing techniques in CR networks.

Sensing technique	Merits	Demerits
Energy detection-based spectrum sensing [5,147–160]	Easy to implement and no prior information of PU is required	Highly prone to noise uncertainty and poor detection at low SNR.
Cyclo-stationary feature-based spectrum sensing [161–166]	Robust to noise uncertainty and reduced false alarm rate at low SNR	Require huge sensing time for optimal performance and high energy consumption.
Matched filter-based spectrum sensing [167–170]	Better sensing and improved detection at lower SNR.	Prior information of PU is required which is usually not available
Covariance-based spectrum sensing [44,171–177]	Prior information of the PU is not required.	Reduced computational complexity
Game theory-based spectrum sensing [180–183]	Optimum resource and power allocation.	Increased computational complexity as the number of users increases.
Machine learning-based spectrum sensing [64,184–194]	Machine learning can learn features from a given data without prior information of the PU and improved sensing time.	Require huge dataset and signal detection is affected by feature selection. The method generates complex hypothetic models.
Neural network-based spectrum sensing [195–198]	Optimal detection performance and can adapt to random environments. Train weights based on historical knowledge from a given data	Effectiveness is based on accurate feature extraction
DL-based spectrum sensing [199–202]	Higher signal detection at low SNR and high gain over existing methods	High computational complexity due to a large number of hidden layers, although this limitation can be curb using GPU/TPU rather than using CPU

Energy detection: The energy detection-based spectrum sensing [5,147–160], has no prior information or knowledge of the PU and thus presents an advantage of the simpler mode of implementation. However, the method is susceptible to noise uncertainty and demonstrates poor detection performance at low SNR.

Cyclo-stationary feature-based detection: In contrast to energy detection, the cyclo-stationary feature-based detection [161–166] is robust to noise uncertainty and reduces the false alarm rate at low SNR. However, this method often requires a huge sensing time for optimal system performance and has a drawback of high energy consumption. Moreover, this method cannot adapt to large spectrum space (complex signals). The cyclostationary feature-based detection provides a better trade-off concerning complexity, implementation, and performance when compared with either matched filter or energy detection-based spectrum sensing.

Matched filter-based spectrum sensing [167–170] presents an improved sensing potential compared with the previously described methods while reporting an improved detection performance at low SNR. However, matched filter-based spectrum sensing has previously shown a shortcoming of over-dependency of prior information from the PU signal. Moreover, this method is not reliable in the presence of PU emulation attacks that pose as a malicious user (impersonates the PU signal).

Another advanced detection technique is covariance-based spectrum sensing [44,171–177]. This method has an improved performance similar to cyclostationary and matched filter detection techniques. However, the high computational complexity renders it unsuitable for practical implementation owing to excessive computations of Fast Fourier Transforms and periodograms [178].

In recent times, artificial intelligence-based spectrum sensing [179] has emerged as a better option compared with conventional spectrum sensing techniques. The success of the artificial intelligence-based techniques could be attributed to extracting and learning informative feature representations from a given data without prior knowledge of the PU.

Game theory-based spectrum sensing [180–183] is employed for optimizing resource allocation and power allocation to CR users. However, due to the increasing number of users, the system becomes computationally complex.

Machine learning-based spectrum sensing [64,184–194] can learn features from a given data without prior information of the PU and has the potential to improve the sensing latency. However, some of the ML methods generate a highly complex hypothetic model and often reliant on a massive amount of datasets, to create a model with proper generalization potential.

Neural network-based spectrum sensing has been investigated in the research papers by [195–198], the studies explored shallow-depth artificial neural networks such as multilayer perceptrons with one or a limited number of hidden layers. This method often yields optimal detection performance and can co-adapt to random environments. The mode of model generation relies on optimizing the predictive error concerning some weighted parameters while updating an optimal weight after several iterative processes. The described learning scheme can be actualized via back-propagation.

Deep Learning (DL) based spectrum sensing is an extension of the MLP with more amount of neural network layers within the hidden layer; an example of some of the deep learning techniques includes; CNN, LSTM, CNN+LSTM, and convolutional long-short-term-memory deep neural networks (CLDNN) [199–202] have shown higher signal detection at lower SNR value and higher gain compared to other approaches. Nevertheless, computation becomes complex with increasing layers. Such computational complexity experienced when training a model on a CPU is often curbed using graphical processing unit (GPU) and tensor processing unit (TPU)".

6. Potential challenges for SS with deep RL

The Deep RL framework involves large volumes of the dataset for optimal system assessment. The availability of such a huge dataset is not feasible for CR application since a reference pool of data like other DL scenarios, as computer vision is not accessible. Most of the current research on the subject depends solely on computer-generated data that challenges the validity in real-world systems. The computer-generated information is normally created using a specified stochastic model that represents the simplified version of a practical scenario, which might overlook some concealed structures. Thus, an improved method of creating a model dataset is essential to validate Deep RL with the real-world scenario [203].

Most of the current studies consider the composition of transmission control, networking, caching, and offloading decisions under a single Deep RL framework to obtain the optimum policy

[204–211]. Nevertheless, from a real point of view, the network structure must pay substantial costs for gathering information. Usually, the cost may be due to high energy consumption, long delay, asynchronous information pre-processing, decreased learning speed, etc. Thus, an open problem is to seek an optimum balance between learning performance and information quality. This enables the agent not to consume excessive resources just to achieve a minimal insignificant increase in the learning performance [203].

Despite the signs of progress made for SS in CR networks, several issues still exist, specifically in situations under attacks, for example, the study by [212] explored spectrum sensing data falsification as a scenario when a malicious user decides to give a false sensing result to deceive the fusion center (FC). These attacks may reduce the detection performance of the CR networks significantly. Hence, it has become imperative to develop more robust multi-agent sensing methods. Detecting PU signals can also come under attacks, such as PU emulation [213]. The task is to achieve an accurate detection while accommodating such attacks that can compromise the detection process in CR networks. In PU emulation, a mischievous user emulates the features of a PU to avoid other SUs from accessing the radio channels. Such security challenges have prompted the necessity for efficient and robust schemes that can guarantee fairness among CR users [214].

7. Conclusion and future research directions

This paper presents a comprehensive overview of Deep RL for spectrum sensing in CR networks. We propose a mathematical hypothetic model of Deep RL-based cooperative spectrum sensing for optimal detection performance. Furthermore, this study reviews the basic principle and characteristics of the various Deep RL algorithms. We have also discussed the advantages and drawbacks of the different techniques that could be adopted for SS in CR networks. Finally, the study present insights into the challenges for the implementation of Deep RL for SS in CR and identify future research directions that remain unresolved and are worth exploring with Deep RL. The discussion will be of interest to a wide range of audiences in telecommunication and artificial intelligence.

Future research work may comprise the following: Spectrum sensing using Deep RL is achieved through the learning and training of agents. It is in contrast with the conventional spectrum sensing schemes that achieve spectrum sensing by designing alternative or iterative algorithms. As a result, further theoretical insights on the working principle of spectrum sensing via Deep RL should be gainfully explored. If the analytical and theoretical framework is established, more Deep RL algorithms may be applied for SS in CR networks.

Optimization is, without a doubt, a crucial part of Deep RL problems. Thus, any advancement in optimization techniques can pave the way to more effective Deep RL algorithms. More recently, there has been much research on the design of optimization algorithms to address complex problems such as non-convex and non-smooth optimizations issues for distributed multi-agent systems [215–217]. However, these algorithms are lacking in the literature of Deep RL algorithms. Furthermore, another open issue that is yet to be resolved is by what means deep architectures may help Deep RL models to transfer learning (transfer knowledge). Particularly, how to employ learned features through the deep neural networks for performing different actions, without altering the real network architectures. We also anticipate seeing notable Deep RL algorithms moving along meta-learning with their applications in radio spectrum management where prior knowledge (for example, active pre-trained networks) could be embedded to improve training time and system performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors wish to thank the Department of Electronics and Telecommunications Engineering, Ahmadu Bello University, Zaria, Nigeria for providing an excellent research environment to complete this work.

References

[1] S. Force, Spectrum policy task force report, in: Federal Communications Commission ET Docket 02, vol. 135, 2002.
[2] S. Haykin, Cognitive radio: brain-empowered wireless communications, IEEE J. Sel. Areas Commun. 23 (2005) 201–220.
[3] I.F. Akyildiz, W.-Y. Lee, M.C. Vuran, S. Mohanty, NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey, Comput. Netw. 50 (2006) 2127–2159.
[4] Z. Han, K. Liu, Resource Allocation for Wireless Networks: Basics, Techniques, and Applications, Cambridge University Press, 2008.
[5] J. Mitola, G.Q. Maguire, Cognitive radio: making software radios more personal, IEEE Pers. Commun. 6 (1999) 13–18.
[6] S. Haykin, Cognitive radio: brain-empowered wireless communications, IEEE J. Sel. Areas Commun. 23 (2005) 201–220.
[7] I.F. Akyildiz, W.-Y. Lee, M.C. Vuran, S. Mohanty, A survey on spectrum management in cognitive radio networks, IEEE Commun. Mag. 46 (2008) 40–48.
[8] S. Kandeepan, A. Giorgetti, Cognitive Radios and Enabling Techniques, Artech House Publishers, Boston, USA, 2012.
[9] P. Pawelczak, K. Nolan, L. Doyle, S.W. Oh, D. Cabric, Cognitive radio: ten years of experimentation and development, IEEE Commun. Mag. 49 (2011) 90–100.
[10] D. Finn, J.C. Tallon, L.A. DaSilva, P. Van Wesemael, S. Pollin, W. Liu, et al., Experimental assessment of tradeoffs among spectrumsensing platforms, in: Proceedings of the 6th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, 2011, pp. 67–74.
[11] W.-Y. Lee, I.F. Akyldiz, A spectrum decision framework for cognitive radio networks, IEEE Trans. Mob. Comput. 10 (2011) 161–174.
[12] Y. Pei, Y.-C. Liang, K.C. Teh, K.H. Li, How much time is needed for qideband spectrum sensing?, IEEE Trans. Wirel. Commun. 8 (2009) 5466–5471.
[13] H. Sun, A. Nallanathan, C.-X. Wang, Y. Chen, Wideband spectrum sensing for cognitive radio networks: a survey, IEEE Wirel. Commun. 20 (2013) 74–81.
[14] E. Axell, G. Leus, E.G. Larsson, H.V. Poor, Spectrum sensing for cognitive radio: state-of-the-art and recent advances, IEEE Signal Process. Mag. 29 (2012) 101–116.
[15] T.W. Ban, W. Choi, B.C. Jung, D.K. Sung, Multi-user diversity in a spectrum sharing system, IEEE Trans. Wirel. Commun. 8 (2009) 102–106.
[16] S. Srinivasa, S.A. Jafar, Cognitive radios for dynamic spectrum access-the throughput potential of cognitive radio: a theoretical perspective, IEEE Commun. Mag. 45 (2007) 73–79.
[17] A. Ghosh, W. Hamouda, On the performance of interference-aware cognitive ad-hoc networks, IEEE Commun. Lett. 17 (2013) 1952–1955.
[18] A. Jafar, S. Srinivasa, The throughput potential of cognitive radio, IEEE Commun. Mag. 45 (2007) 73–79.
[19] N. Devroye, P. Mitran, V. Tarokh, Achievable rates in cognitive radio channels, IEEE Trans. Inf. Theory 52 (2006) 1813–1827.
[20] A. Jovicic, P. Viswanath, Cognitive radio: an information-theoretic perspective, IEEE Trans. Inf. Theory 55 (2009) 3945–3958.
[21] A.K. Mishra, D.L. Johnson, White Space Communication: Advances, Developments and Engineering Challenges, Springer, 2014.
[22] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[23] T. Yucek, H. Arslan, A survey of spectrum sensing algorithms for cognitive radio applications, IEEE Commun. Surv. Tutor. 11 (2009) 116–130.
[24] W.D. Horne, Adaptive spectrum access: using the full spectrum space, in: Proc. Telecommunications Policy Research Conference, TPRC, 2003.
[25] S. Haykin, D.J. Thomson, J.H. Reed, Spectrum sensing for cognitive radio, Proc. IEEE 97 (2009) 849–877.
[26] C. Cormio, K.R. Chowdhury, A survey on MAC protocols for cognitive radio networks, Ad Hoc Netw. 7 (2009) 1315–1329.
[27] B. Shen, K.S. Kwak, Soft combination schemes for cooperative spectrum sensing in cognitive radio networks, ETRI J. 31 (2009) 263–270.
[28] M. Kam, Q. Zhu, W.S. Gray, Optimal data fusion of correlated local decisions in multiple sensor detection systems, IEEE Trans. Aerosp. Electron. Syst. 28 (1992) 916–920.
[29] B. Chen, R. Jiang, T. Kasetkasem, P.K. Varshney, Channel aware decision fusion in wireless sensor networks, IEEE Trans. Signal Process. 52 (2004) 3454–3458.

[30] Z. Chair, P. Varshney, Optimal data fusion in multiple sensor detection systems, IEEE Trans. Aerosp. Electron. Syst. (1986) 98–101.

[31] P. Verma, B. Singh, On the decision fusion for cooperative spectrum sensing in cognitive radio networks, Wirel. Netw. 23 (2017) 2253–2262.

[32] T. Yucek, H. Arslan, A survey of spectrum sensing algorithms for cognitive radio applications, IEEE Commun. Surv. Tutor. 11 (2009) 116–130.

[33] H. Poor, An Introduction to Signal Detection and Estimation, Dowden and Culver, 1994.

[34] S.M. Kay, Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory, Signal Processing, Prentice Hall, Upper Saddle River, NJ, 1998.

[35] F.F. Digham, M.-S. Alouini, M.K. Simon, On the energy detection of unknown signals over fading channels, in: IEEE International Conference on Communications, 2003 ICC'03, 2003, pp. 3575–3579.

[36] N. Yadav, S. Rathi, A comprehensive study of spectrum sensing techniques in cognitive radio, Int. J. Adv. Eng. Technol. (2011) 1963–2231.

[37] A. Fanan, N. Riley, M. Mehdawi, M. Ammar, M. Zolfaghari, Survey: a comparison of spectrum sensing techniques in cognitive radio, in: Int'l Conference on Image Processing, Computers and Industrial Engineering (ICICIE'2014), Conference Proceedings, Jan. 15–16, 2014, Kuala Lumpur (Malaysia), ISBN 978-93-82242-67-3, 2014, p. 65.

[38] Z. Tabakovic, A Survey of Cognitive Radio Systems, Croatian Post and Electronic Communications Agency, 2013.

[39] I.K. Aulakh, Spectrum Sensing for Wireless Communication Networks, IEEE, 2009.

[40] A. Singh, V. Saxena, Different spectrum sensing techniques used in non-cooperative system, Int. J. Eng. Innov. Technol. 1 (2012).

[41] S. Ziafat, W. Ejaz, H. Jamal, Spectrum sensing techniques for cognitive radio networks: performance analysis, in: IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals, IMWS-IRFPT, 2011, 2011, pp. 1–4.

[42] K. Sithamparanathan, A. Giorgetti, Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation, and Localization, Artech House, 2012.

[43] H. Urkowitz, Energy detection of unknown deterministic signals, Proc. IEEE 55 (1967) 523–531.

[44] Y. Zeng, Y.-C. Liang, Eigenvalue-based spectrum sensing algorithms for cognitive radio, IEEE Trans. Commun. 57 (2009) 1784–1793.

[45] V. Prithiviraj, B. Sarankumar, A. Kalaiyarasan, P.P. Chandru, N.N. Singh, Cyclostationary analysis method of spectrum sensing for cognitive radio, in: 2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, Wireless VITAE, 2011, pp. 1–5.

[46] A. Nasser, A. Mansour, K.C. Yao, H. Charara, M. Chaitou, Efficient spectrum sensing approaches based on waveform detection, in: The Third International Conference on e-Technologies and Networks for Development, ICeND2014, 2014, pp. 13–17.

[47] W.A. Gardner, Introduction to Random Processes with Applications to Signals and Systems, MacMillan Co., New York, 1986, p. 447.

[48] W.A. Gardner, An introduction to cyclostationary signals, in: Cyclostationarity in Communications and Signal Processing, IEEE Press, New York, 1994, pp. 1–90.

[49] C. Tom, Investigation and Implementation of Computationally-Efficient Algorithm for Cyclic Spectral Analysis, Carleton University, 1995.

[50] W.M. Jang, Blind cyclostationary spectrum sensing in cognitive radios, IEEE Commun. Lett. 18 (2014) 393–396.

[51] A. Nasser, A. Mansour, K.C. Yao, H. Abdallah, Spectrum sensing for half and full-duplex cognitive radio, in: Spectrum Access and Management for Cognitive Radio Networks, Springer, 2017, pp. 15–50.

[52] D. Cohen, Y.C. Eldar, Sub-Nyquist cyclostationary detection for cognitive radio, IEEE Trans. Signal Process. 65 (2017) 3004–3019.

[53] S.K. Sharma, T.E. Bogale, S. Chatzinotas, B. Ottersten, L.B. Le, X. Wang, Cognitive radio techniques under practical imperfections: a survey, IEEE Commun. Surv. Tutor. 17 (2015) 1858–1884.

[54] M. Jin, Q. Guo, J. Xi, Y. Li, Y. Li, On spectrum sensing of OFDM signals at low SNR: new detectors and asymptotic performance, IEEE Trans. Signal Process. 65 (2017) 3218–3233.

[55] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2015) 1153–1176.

[56] Z. Han, R. Zheng, H.V. Poor, Repeated auctions with Bayesian nonparametric learning for spectrum access in cognitive radio networks, IEEE Trans. Wirel. Commun. 10 (2011) 890–900.

[57] J. Lundén, V. Koivunen, S.R. Kulkarni, H.V. Poor, Reinforcement learning based distributed multiagent sensing policy for cognitive radio networks, in: 2011 IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN, 2011, pp. 642–646.

[58] M. Bkassiny, S.K. Jayaweera, K.A. Avery, Distributed reinforcement learning based MAC protocols for autonomous cognitive secondary users, in: 2011 20th Annual Wireless and Optical Communications Conference, WOCC, 2011, pp. 1–6.

[59] A. Galindo-Serrano, L. Giupponi, Distributed Q-learning for aggregated interference control in cognitive radio networks, IEEE Trans. Veh. Technol. 59 (2010) 1823–1834.

[60] Y.B. Reddy, Detecting primary signals for efficient utilization of spectrum using Q-learning, in: Fifth International Conference on Information Technology: New Generations, ITNG 2008, 2008, pp. 360–365.

[61] Q. Zhu, Z. Han, T. Başar, No-regret learning in collaborative spectrum sensing with malicious nodes, in: 2010 IEEE International Conference on Communications, 2010, pp. 1–6.

[62] K.M. Thilina, K.W. Choi, N. Saquib, E. Hossain, Pattern classification techniques for cooperative spectrum sensing in cognitive radio networks: SVM and W-KNN approaches, in: 2012 IEEE Global Communications Conference, GLOBECOM, 2012, pp. 1260–1265.

[63] M. Tang, Z. Zheng, G. Ding, Z. Xue, Efficient TV white space database construction via spectrum sensing and spatial inference, in: 2015 IEEE 34th International Performance Computing and Communications Conference, IPCCC, 2015, pp. 1–5.

[64] A.M. Mikaeil, B. Guo, Z. Wang, Machine learning to data fusion approach for cooperative spectrum sensing, in: 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2014, pp. 429–434.

[65] Z. Li, W. Wu, X. Liu, P. Qi, Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks, IET Commun. 12 (2018) 2485–2492.

[66] Z. Liu, C. Li, X. Gao, G. Wang, J. Yang, Ensemble-based depression detection in speech, in: 2017 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, 2017, pp. 975–980.

[67] V. Timčenko, S. Gajin, Ensemble classifiers for supervised anomaly based network intrusion detection, in: 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing, ICCP, 2017, pp. 13–19.

[68] T. Guo, P. Papadopoulos, P. Mohammed, J. Milanovic, Comparison of ensemble decision tree methods for on-line identification of power system dynamic signature considering availability of PMU measurements, in: 2015 IEEE Eindhoven PowerTech, 2015, pp. 1–6.

[69] J. Moeyersons, C. Varon, D. Testelmans, B. Buyse, S. Van Huffel, Ecg artefact detection using ensemble decision trees, in: 2017 Computing in Cardiology, CinC, 2017, pp. 1–4.

[70] Z. Wei, P. Zhang, Empirical study of pedestrian detection algorithm based on ensemble learning, in: 2017 IEEE 13th International Symposium on Autonomous Decentralized System, ISADS, 2017, pp. 175–180.

[71] S.S. Madani, A. Abbaspour, M. Beiraghi, P.Z. Dehkordi, A.M. Ranjbar, Islanding detection for PV and DFIG using decision tree and AdaBoost algorithm, in: 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe, ISGT Europe, 2012, pp. 1–8.

[72] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198.

[73] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (2006) 21–45.

[74] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (2010) 1–39.

[75] B.F. Lo, I.F. Akyildiz, Reinforcement learning-based cooperative sensing in cognitive radio ad hoc networks, in: 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2010, pp. 2244–2249.

[76] S. Vakili, K. Liu, Q. Zhao, Deterministic sequencing of exploration and exploitation for multi-armed bandit problems, IEEE J. Sel. Top. Signal Process. 7 (2013) 759–767.

[77] Y. Kawaguchi, M. Togami, Adaptive boolean compressive sensing by using multi-armed bandit, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2016, pp. 3261–3265.

[78] Q. Zhu, Z. Han, T. Basar, No-regret learning in collaborative spectrum sensing with malicious nodes, in: 2010 IEEE International Conference on Communications, 2010, pp. 1–6.

[79] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, H. Li, Intelligent power control for spectrum sharing in cognitive radios: a deep reinforcement learning approach, IEEE Access 6 (2018) 25463–25473.

[80] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, L. Liu, Distributive dynamic spectrum access through deep reinforcement learning: a reservoir computing-based approach, IEEE Int. Things J. 6 (2018) 1938–1948.

[81] V.Q. Do, I. Koo, Learning frameworks for cooperative spectrum sensing and energy-efficient data protection in cognitive radio networks, Appl. Sci. 8 (2018) 722.

[82] P. Yang, L. Li, J. Yin, H. Zhang, W. Liang, W. Chen, et al., Dynamic spectrum access in cognitive radio networks using deep reinforcement learning and evolutionary game, in: 2018 IEEE/CIC International Conference on Communications in China, ICCC, 2018, pp. 405–409.

[83] S. Zheng, S. Chen, P. Qi, H. Zhou, X. Yang, Spectrum sensing based on deep learning classification for cognitive radios, China Commun. 17 (2020) 138–148.

[84] Y.-J. Li, H.-Y. Chang, Y.-J. Lin, P.-W. Wu, Y.-C. FrankWang, Deep reinforcement learning for playing 2.5 D fighting games, in: 2018 25th IEEE International Conference on Image Processing, ICIP, 2018, pp. 3778–3782.

[85] D.P. Bertsekas, Reinforcement Learning and Optimal Control, Athena Scientific, 2019.

[86] Z. Zhang, D. Zhang, R.C. Qiu, Deep reinforcement learning for power system applications: an overview, CSEE J. Power Energy Syst. 6 (2019) 213–225.

[87] O. Sigaud, O. Buffet, Markov Decision Processes in Artificial Intelligence, John Wiley & Sons, 2013.

[88] P. Macaluso, Deep Reinforcement Learning for Autonomous Systems, Politecnico di Torino, 2020.

[89] V. François-Lavet, P. Henderson, R. Islam, M.G. Bellemare, J. Pineau, An introduction to deep reinforcement learning, Found. Trends Mach. Learn. 11 (2018) 219–354.

[90] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[91] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, 1996.

[92] S.P. Singh, R.S. Sutton, Reinforcement learning with replacing eligibility traces, Mach. Learn. 22 (1996) 123–158.

[93] A.R. Cassandra, Exact and Approximate Algorithms for Partially Observable Markov Decision Processes, 1998.

[94] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.

[95] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[96] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for Boltzmann machines, Cogn. Sci. 9 (1985) 147–169.

[97] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: ICML, 2010.

[98] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, et al., A state-of-the-art survey on deep learning theory and architectures, Electronics 8 (2019) 292.

[99] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[100] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[101] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (1990) 179–211.

[102] M.I. Jordan, Serial order: a parallel distributed processing approach, in: Advances in Psychology, vol. 121, Elsevier, 1997, pp. 471–495.

[103] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.

[104] C. Olah, Understanding Lstm Networks, 2015.

[105] X.-H. Le, H.V. Ho, G. Lee, S. Jung, Application of long short-term memory (LSTM) neural network for flood forecasting, Water 11 (2019) 1387.

[106] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, preprint, arXiv:1412.3555, 2014.

[107] Z. Zhang, D. Zhang, R.C. Qiu, Deep reinforcement learning for power system: an overview, CSEE J. Power Energy Syst. (2019).

[108] C. Watkins, Learning from delayed rewards, a PhD thesis at King's College, Cambridge, England, 1989.

[109] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.

[110] M. Sewak, Deep Reinforcement Learning: Frontiers of Artificial Intelligence, Springer, 2019.

[111] L.-J. Lin, Self-improving reactive agents based on reinforcement learning, planning and teaching, Mach. Learn. 8 (1992) 293–321.

[112] V.R. Konda, J.N. Tsitsiklis, Actor-critic algorithms, in: Advances in Neural Information Processing Systems, 2000, pp. 1008–1014.

[113] D. Precup, Eligibility traces for off-policy policy evaluation, in: Computer Science Department Faculty Publication Series, 2000, p. 80.

[114] R. Munos, T. Stepleton, A. Harutyunyan, M. Bellemare, Safe and efficient off-policy reinforcement learning, in: Advances in Neural Information Processing Systems, 2016, pp. 1054–1062.

[115] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, et al., Sample efficient actor-critic with experience replay, preprint, arXiv:1611.01224, 2016.

[116] A. Gruslys, M.G. Azar, M.G. Bellemare, R. Munos, The reactor: a sample-efficient actor-critic architecture, preprint, arXiv:1704.04651, 2017.

[117] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, et al., Asynchronous methods for deep reinforcement learning, in: International Conference on Machine Learning, 2016, pp. 1928–1937.

[118] P. Thomas, Bias in natural actor-critic algorithms, in: International Conference on Machine Learning, 2014, pp. 441–448.

[119] M. Deisenroth, C.E. Rasmussen, PILCO: a model-based and data-efficient approach to policy search, in: Proceedings of the 28th International Conference on Machine Learning, ICML-11, 2011, pp. 465–472.

[120] N. Wahlström, T.B. Schön, M.P. Deisenroth, From pixels to torques: policy learning with deep dynamical models, preprint, arXiv:1502.02251, 2015.

[121] M. Morari, J.H. Lee, Model predictive control: past, present and future, Comput. Chem. Eng. 23 (1999) 667–682.

[122] S. Levine, V. Koltun, Guided policy search, in: International Conference on Machine Learning, 2013, pp. 1–9.

[123] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (2016) 484.

[124] A.T. Nassar, Y. Yilmaz, Reinforcement-learning-based resource allocation in fog radio access networks for various IoT environments, preprint, arXiv:1806.04582, 2018.

[125] Y. Yu, T. Wang, S.C. Liew, Deep-reinforcement learning multiple access for heterogeneous wireless networks, IEEE J. Sel. Areas Commun. 37 (2019) 1277–1290.

[126] S. Wang, H. Liu, P.H. Gomes, B. Krishnamachari, Deep reinforcement learning for dynamic multichannel access in wireless networks, IEEE Trans. Cogn. Commun. Netw. 4 (2018) 257–265.

[127] Y. Zhang, Q. Zhang, B. Cao, P. Chen, Model free dynamic sensing order selection for imperfect sensing multichannel cognitive radio networks: a Q-learning approach, in: 2014 IEEE International Conference on Communication Systems, 2014, pp. 364–368.

[128] O. Naparstek, K. Cohen, Deep multi-user reinforcement learning for distributed dynamic spectrum access, IEEE Trans. Wirel. Commun. 18 (2018) 310–323.

[129] S. Liu, X. Hu, W. Wang, Deep reinforcement learning based dynamic channel allocation algorithm in multibeam satellite systems, IEEE Access 6 (2018) 15733–15742.

[130] H. Li, Multiagent-learning for aloha-like spectrum access in cognitive radio systems, EURASIP J. Wirel. Commun. Netw. 2010 (2010) 1–15.

[131] R. Fan, H. Jiang, Optimal multi-channel cooperative sensing in cognitive radio networks, IEEE Trans. Wirel. Commun. 9 (2010) 1128–1138.

[132] H. Jiang, L. Lai, R. Fan, H.V. Poor, Optimal selection of channel sensing order in cognitive radio, IEEE Trans. Wirel. Commun. 8 (2009) 297–307.

[133] H.T. Cheng, W. Zhuang, Simple channel sensing order in cognitive radio networks, IEEE J. Sel. Areas Commun. 29 (2011) 676–688.

[134] L. Lai, H. El Gamal, H. Jiang, H.V. Poor, Cognitive medium access: exploration, exploitation, and competition, IEEE Trans. Mob. Comput. 10 (2010) 239–253.

[135] Y. Chen, Q. Zhao, A. Swami, Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors, IEEE Trans. Inf. Theory 54 (2008) 2053–2071.

[136] S.H.A. Ahmad, M. Liu, T. Javidi, Q. Zhao, B. Krishnamachari, Optimality of myopic sensing in multichannel opportunistic access, IEEE Trans. Inf. Theory 55 (2009) 4040–4050.

[137] L. Lai, H. Jiang, H.V. Poor, Medium access in cognitive radio networks: a competitive multi-armed bandit framework, in: 2008 42nd Asilomar Conference on Signals, Systems and Computers, 2008, pp. 98–102.

[138] K. Liu, Q. Zhao, Indexability of restless bandit problems and optimality of Whittle index for dynamic multichannel access, IEEE Trans. Inf. Theory 56 (2010) 5547–5567.

[139] K. Liu, Q. Zhao, B. Krishnamachari, Dynamic multichannel access with imperfect channel state detection, IEEE Trans. Signal Process. 58 (2010) 2795–2808.

[140] C. Tekin, M. Liu, Online learning of rested and restless bandits, IEEE Trans. Inf. Theory 58 (2012) 5588–5611.

[141] Q. Zhao, B. Krishnamachari, K. Liu, On myopic sensing for multi-channel opportunistic access: structure, optimality, and performance, IEEE Trans. Wirel. Commun. 7 (2008) 5431–5440.

[142] Q. Zhao, A. Swami, A survey of dynamic spectrum access: signal processing and networking perspectives, in: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, 2007, pp. 1349–1352.

[143] C.-H. Liu, J.A. Tran, P. Pawelczak, D. Cabric, Traffic-aware channel sensing order in dynamic spectrum access networks, IEEE J. Sel. Areas Commun. 31 (2013) 2312–2323.

[144] G. Umashankar, A.P. Kannu, Throughput optimal multi-slot sensing procedure for a cognitive radio, IEEE Commun. Lett. 17 (2013) 2292–2295.

[145] M. Barkat, Signal Detection and Estimation, Artech House, 1991.

[146] D.P. Bertsekas, D.P. Bertsekas, D.P. Bertsekas, D.P. Bertsekas, Dynamic Programming and Optimal Control, vol. 1, Athena Scientific, Belmont, MA, 1995.

[147] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (2015) 2347–2376.

[148] P. Rawat, K.D. Singh, J.M. Bonnin, Cognitive radio for M2M and Internet of Things: a survey, Comput. Commun. 94 (2016) 1–29.

[149] A. Ranjan, B. Singh, Design and analysis of spectrum sensing in cognitive radio based on energy detection, in: 2016 International Conference on Signal and Information Processing, IConSIP, 2016, pp. 1–5.

[150] D.M.M. Plata, Á.G.A. Reátiga, Evaluation of energy detection for spectrum sensing based on the dynamic selection of detection-threshold, Proc. Eng. 35 (2012) 135–143.

[151] S. Atapattu, C. Tellambura, H. Jiang, Energy Detection for Spectrum Sensing in Cognitive Radio, vol. 6, Springer, 2014.

[152] L. Ruan, Y. Li, W. Cheng, Z. Wu, A robust threshold optimization approach for energy detection based spectrum sensing with noise uncertainty, in: 2015 IEEE 10th Conference on Industrial Electronics and Applications, ICIEA, 2015, pp. 161–165.

[153] M.Z. Alom, T.K. Godder, M.N. Morshed, A. Maali, Enhanced spectrum sensing based on energy detection in cognitive radio network using adaptive threshold, in: 2017 International Conference on Networking, Systems and Security, NSysS, 2017, pp. 138–143.
[154] Y. Arjoune, Z. El Mrabet, H. El Ghazi, A. Tamtaoui, Spectrum sensing: enhanced energy detection technique based on noise measurement, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC, 2018, pp. 828–834.
[155] A. Eslami, S. Karamzadeh, Performance analysis of double threshold energy detection-based spectrum sensing in low SNRs over Nakagami-m fading channels with noise uncertainty, in: 2016 24th Signal Processing and Communication Application Conference, SIU, 2016, pp. 309–312.
[156] X. Ling, B. Wu, H. Wen, P.-H. Ho, Z. Bao, L. Pan, Adaptive threshold control for energy detection based spectrum sensing in cognitive radios, IEEE Wirel. Commun. Lett. 1 (2012) 448–451.
[157] Y. Zeng, Y.-C. Liang, R. Zhang, Blindly combined energy detection for spectrum sensing in cognitive radio, IEEE Signal Process. Lett. 15 (2008) 649–652.
[158] A. Muralidharan, P. Venkateswaran, S. Ajay, D.A. Prakash, M. Arora, S. Kirthiga, An adaptive threshold method for energy based spectrum sensing in cognitive radio networks, in: 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT, 2015, pp. 8–11.
[159] M. Sarker, Energy detector based spectrum sensing by adaptive threshold for low SNR in CR networks, in: 2015 24th Wireless and Optical Communication Conference, WOCC, 2015, pp. 118–122.
[160] J. Wu, T. Luo, G. Yue, An energy detection algorithm based on double-threshold in cognitive radio systems, in: 2009 First International Conference on Information Science and Engineering, 2009, pp. 493–496.
[161] P.S. Yawada, A.J. Wei, Cyclostationary detection based on non-cooperative spectrum sensing in cognitive radio network, in: 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, CYBER, 2016, pp. 184–187.
[162] I. Ilyas, S. Paul, A. Rahman, R.K. Kundu, Comparative evaluation of cyclostationary detection based cognitive spectrum sensing, in: 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON, 2016, pp. 1–7.
[163] M.-A. Damavandi, S. Nader-Esfahani, Compressive wideband spectrum sensing in cognitive radio systems based on cyclostationary feature detection, in: 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, 2015, pp. 282–287.
[164] D. Cohen, Y.C. Eldar, Compressed cyclostationary detection for cognitive radio, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2017, pp. 3509–3513.
[165] S.K. Sharma, T.E. Bogale, S. Chatzinotas, L.B. Le, X. Wang, B. Ottersten, Improving robustness of cyclostationary detectors to cyclic frequency mismatch using Slepian basis, in: 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, 2015, pp. 456–460.
[166] H. Reyes, S. Subramaniam, N. Kaabouch, W.C. Hu, A spectrum sensing technique based on autocorrelation and Euclidean distance and its comparison with energy detection for cognitive radio networks, Comput. Electr. Eng. 52 (2016) 319–327.
[167] F. Salahdine, H. El Ghazi, N. Kaabouch, W.F. Fihri, Matched filter detection with dynamic threshold for cognitive radio networks, in: 2015 International Conference on Wireless Networks and Mobile Communications, WINCOM, 2015, pp. 1–6.
[168] X. Zhang, R. Chai, F. Gao, Matched filter based spectrum sensing and power level detection for cognitive radio network, in: 2014 IEEE Global Conference on Signal and Information Processing, GlobalSIP, 2014, pp. 1267–1270.
[169] C. Jiang, Y. Li, W. Bai, Y. Yang, J. Hu, Statistical matched filter based robust spectrum sensing in noise uncertainty environment, in: 2012 IEEE 14th International Conference on Communication Technology, 2012, pp. 1209–1213.
[170] Q. Lv, F. Gao, Matched filter based spectrum sensing and power level recognition with multiple antennas, in: 2015 IEEE China Summit and International Conference on Signal and Information Processing, ChinaSIP, 2015, pp. 305–309.
[171] K.S. Kumar, R. Saravanan, R. Muthaiah, Cognitive radio spectrum sensing algorithms based on eigenvalue and covariance methods, Int. J. Eng. Technol. 5 (2013) 385–395.
[172] Y. Zeng, Y.-C. Liang, Covariance based signal detections for cognitive radio, in: 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2007, pp. 202–207.
[173] Y. Zeng, Y.-C. Liang, Maximum-minimum eigenvalue detection for cognitive radio, in: 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007, pp. 1–5.
[174] Y. Zeng, Y.-C. Liang, Spectrum-sensing algorithms for cognitive radio based on statistical covariances, IEEE Trans. Veh. Technol. 58 (2008) 1804–1815.
[175] Q. Zhang, Advanced detection techniques for cognitive radio, in: 2009 IEEE International Conference on Communications, 2009, pp. 1–5.
[176] B. Zayen, A. Hayar, K. Kansanen, Blind spectrum sensing for cognitive radio based on signal space dimension estimation, in: 2009 IEEE International Conference on Communications, 2009, pp. 1–5.
[177] Y. Zeng, C.L. Koh, Y.-C. Liang, Maximum eigenvalue detection: theory and application, in: 2008 IEEE International Conference on Communications, 2008, pp. 4160–4164.
[178] P. Sivagurunathan, P. Ramakrishnan, N. Sathishkumar, Recent paradigms for efficient spectrum sensing in cognitive radio networks: issues and challenges, J. Phys. Conf. Ser. 1717 (2021) 012057.
[179] N. Abbas, Y. Nasser, K. El Ahmad, Recent advances on artificial intelligence and learning techniques in cognitive radio networks, EURASIP J. Wirel. Commun. Netw. 2015 (2015) 1–20.
[180] B. Wang, Y. Wu, K.R. Liu, Game theory for cognitive radio networks: an overview, Comput. Netw. 54 (2010) 2537–2561.
[181] J. Gupta, P. Chauhan, M. Nath, M. Manvithasree, S.K. Deka, N. Sarma, Coalitional game theory based cooperative spectrum sensing in CRNs, in: Proceedings of the 18th International Conference on Distributed Computing and Networking, 2017, pp. 1–7.
[182] S. Salim, S. Moh, An energy-efficient game-theory-based spectrum decision scheme for cognitive radio sensor networks, Sensors 16 (2016) 1009.
[183] A.A. Anghuwo, Y. Liu, X. Tan, S. Liu, Spectrum allocation based on game theory in cognitive radio networks, in: International Symposium on Information and Automation, 2010, pp. 1–9.
[184] V. Balaji, P. Kabra, P. Saieesh, C. Hota, G. Raghurama, Cooperative spectrum sensing in cognitive radios using perceptron learning for IEEE 802.22 wran, Proc. Comput. Sci. 54 (2015) 14–23.
[185] K. Zhang, J. Li, F. Gao, Machine learning techniques for spectrum sensing when primary user has multiple transmit powers, in: 2014 IEEE International Conference on Communication Systems, 2014, pp. 137–141.
[186] B. Khalfi, A. Zaid, B. Hamdaoui, When machine learning meets compressive sampling for wideband spectrum sensing, in: 2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC, 2017, pp. 1120–1125.
[187] Y. Lu, P. Zhu, D. Wang, M. Fattouche, Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks, in: 2016 IEEE Wireless Communications and Networking Conference, 2016, pp. 1–6.
[188] D. Wang, Z. Yang, An novel spectrum sensing scheme combined with machine learning, in: 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI, 2016, pp. 1293–1297.
[189] E. Ghazizadeh, B. Nikpour, D.A. Moghadam, H. Nezamabadi-pour, A PSO-based weighting method to enhance machine learning techniques for cooperative spectrum sensing in CR networks, in: 2016 1st Conference on Swarm Intelligence and Evolutionary Computation, CSIEC, 2016, pp. 113–118.
[190] G. Ding, Q. Wu, Y.-D. Yao, J. Wang, Y. Chen, Kernel-based learning for statistical signal processing in cognitive radio networks: theoretical foundations, example applications, and future directions, IEEE Signal Process. Mag. 30 (2013) 126–136.
[191] Y. Li, Q. Peng, Achieving secure spectrum sensing in presence of malicious attacks utilizing unsupervised machine learning, in: MILCOM 2016-2016 IEEE Military Communications Conference, 2016, pp. 174–179.
[192] G. Nie, G. Ding, L. Zhang, Q. Wu, Byzantine defense in collaborative spectrum sensing via Bayesian learning, IEEE Access 5 (2017) 20089–20098.
[193] F. Farmani, M. Abbasi-Jannatabad, R. Berangi, Detection of SSDF attack using SVDD algorithm in cognitive radio networks, in: 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks, 2011, pp. 201–204.
[194] H.A. Shah, I. Koo, Reliable Machine Learning Based Spectrum Sensing in Cognitive Radio Networks, Wireless Communications and Mobile Computing, vol. 2018, 2018.
[195] B. Varatharajana, E. Praveen, E. Vinotha, Neural network aided enhanced spectrum sensing in cognitive radio, Proc. Eng. 38 (2012) 82–88.
[196] M.R. Vyas, D.K. Patel, M. Lopez-Benitez, Artificial neural network based hybrid spectrum sensing scheme for cognitive radio, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, 2017, pp. 1–7.
[197] Y.-J. Tang, Q.-Y. Zhang, W. Lin, Artificial neural network based spectrum sensing method for cognitive radio, in: 2010 6th International Conference on Wireless Communications Networking and Mobile Computing, WiCOM, 2010, pp. 1–4.
[198] Y. Lee, I. Koo, A neural network-based cooperative spectrum sensing scheme for cognitive radio systems, in: International Conference on Intelligent Computing, 2010, pp. 364–371.
[199] K. Yang, Z. Huang, X. Wang, X. Li, A blind spectrum sensing method based on deep learning, Sensors 19 (2019) 2270.
[200] J. Gao, X. Yi, C. Zhong, X. Chen, Z. Zhang, Deep learning for spectrum sensing, IEEE Wirel. Commun. Lett. 8 (2019) 1727–1730.
[201] W. Lee, M. Kim, D.-H. Cho, Deep cooperative sensing: cooperative spectrum sensing based on convolutional neural networks, IEEE Trans. Veh. Technol. 68 (2019) 3005–3009.
[202] Q. Cheng, Z. Shi, D.N. Nguyen, E. Dutkiewicz, Deep learning network based spectrum sensing methods for OFDM systems, preprint, arXiv:1807.09414, 2018.

[203] N.C. Luong, D.T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, et al., Applications of deep reinforcement learning in communications and networking: a survey, IEEE Commun. Surv. Tutor. 21 (2019) 3133–3174.
[204] Y. He, Z. Zhang, Y. Zhang, A big data deep reinforcement learning approach to next generation green wireless networks, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, 2017, pp. 1–6.
[205] Y. He, C. Liang, Z. Zhang, F.R. Yu, N. Zhao, H. Yin, et al., Resource allocation in software-defined and information-centric vehicular networks with mobile edge computing, in: 2017 IEEE 86th Vehicular Technology Conference, VTC-Fall, 2017, pp. 1–5.
[206] Y. He, F.R. Yu, N. Zhao, H. Yin, A. Boukerche, Deep reinforcement learning (DRL)-based resource management in software-defined and virtualized vehicular ad hoc networks, in: Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, 2017, pp. 47–54.
[207] Y. He, N. Zhao, H. Yin, Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach, IEEE Trans. Veh. Technol. 67 (2017) 44–55.
[208] R.Q. Hu, Mobility-aware edge caching and computing in vehicle networks: a deep reinforcement learning, IEEE Trans. Veh. Technol. 67 (2018) 10190–10203.
[209] Y. He, F.R. Yu, N. Zhao, V.C. Leung, H. Yin, Software-defined networks with mobile edge computing and caching for smart cities: a big data deep reinforcement learning approach, IEEE Commun. Mag. 55 (2017) 31–37.
[210] Y. He, F.R. Yu, N. Zhao, H. Yin, Secure social networks in 5G systems with mobile edge computing, caching, and device-to-device communications, IEEE Wirel. Commun. 25 (2018) 103–109.
[211] Y. He, C. Liang, R. Yu, Z. Han, Trust-based social networks with computing, caching and communications: a deep reinforcement learning approach, IEEE Trans. Netw. Sci. Eng. (2018).
[212] M.R. Manesh, N. Kaabouch, Security threats and countermeasures of MAC layer in cognitive radio networks, Ad Hoc Netw. 70 (2018) 85–102.
[213] W.F. Fihri, Y. Arjoune, H. El Ghazi, N. Kaabouch, B. Abou El Majd, A particle swarm optimization based algorithm for primary user emulation attack detection, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC, 2018, pp. 823–827.
[214] Y. Arjoune, Z.E. Mrabet, N. Kaabouch, Multi-attributes, utility-based, channel quality ranking mechanism for cognitive radio networks, Appl. Sci. 8 (2018) 628.
[215] P. Bianchi, J. Jakubowicz, Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization, IEEE Trans. Autom. Control 58 (2012) 391–405.
[216] P. Di Lorenzo, G. Scutari, Next: in-network nonconvex optimization, IEEE Trans. Signal Inf. Process. Netw. 2 (2016) 120–136.
[217] M. Hong, D. Hajinezhad, M.-M. Zhao, Prox-PDA: the proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks, in: Proceedings of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 1529–1538.

Felix Obite received his Postgraduate Diploma (PGD) in Electronics & Telecommunication Engineering from Ahmadu Bello University, Zaria, Nigeria in 2012. He obtained his Master of Engineering Degree in Electrical Electronics and Telecommunication from the prestigious Universiti Teknologi Malaysia in 2017. He is currently undergoing his Ph.D. degree in Telecommunications Engineering at Ahmadu Bello University, Zaria, Nigeria. He has research interests in cognitive radio, artificial intelligence, computer vision, 5G massive MIMO systems, nanoelectronic devices (carbon nanotubes & graphene), biosensors and applications, and optical access networks.

Aliyu D. Usman received his Master of Engineering Degree from Bayero University Kano, Nigeria in 2006, and a Ph.D. degree in Biomedical and Microwave Engineering from the prestigious Universiti Putra Malaysia (UPM) in 2011. He is currently an Associate Professor of Telecommunication Engineering and the Head of the Department of Electronics and Telecommunications Engineering, Ahmadu Bello University, Zaria, Nigeria. He has more than 100 National and International publications from reputable journals and conferences. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), Nigerian Society of Engineers (MNSE), Registered Engineer with Council for Regulation of Engineering Practice in Nigeria (COREN). His research interest is in Antenna Technology, WSN, Wireless Communications, RF-EMF Effect, applications of Artificial Intelligence to communications networks, and Terahertz frequencies.

Emmanuel Okafor is a Lecturer in the field of machine learning, computer vision, and control engineering at the Department of Computer Engineering, Ahmadu Bello University (ABU), Nigeria. He earned Bachelor's Degree in Electrical Engineering (2010) and a Master's Degree in Control Engineering (2014) from ABU. Dr. Okafor obtained a Ph.D. in the field of Artificial Intelligence (2019) from the University of Groningen, the Netherlands. His current research interests include computer vision, deep learning, time series forecasting, reinforcement learning, robotics, and optimization.