EAI Endorsed Transactions
on Mobile Communications and Applications
Research Article EAI.EU

# Cooperative Spectrum Sensing using DQN in CRN

M. Moneesh[1,*], T. Sai Tejaswi[1], T. Sai Yeshwanth[1], M. Sai Harshitha[1], G. Chakravarthy[2]

[1]Department of ECE, V R Siddhartha engineering college, Vijayawada, India.
[2]Assistant professor, Department of ECE, V R Siddhartha engineering college, Vijayawada, India.

## Abstract

Abstract —It is imperative to address the problem of spectrum under usage and inefficiency because of the increasing spectrum demand and slender spectrum resources. One of the salient functions of cognitive radio is spectrum sensing which is used to avoid the interference of the unlicensed secondary users with licensed primary users and spot the available spectrum for enhancing the spectrum usage. The frequency band that a secondary user can utilize without interfering with any licensed primary users are called spectrum holes. Cooperative sensing is a remedy to improve the sensing performance, in which secondary users (SUs) cooperate among themselves to sense the spectrum and find the spectrum holes. Here we propose a deep reinforcement learning based spectrum sensing to discover the spectrum holes. We implement a deep reinforcement learning based method called Deep Q-Network (DQN) to find the spectrum holes. The secondary users (SU) uses the DQN to find the vacant channels in the spectrum effectively. The secondary user (SU) senses the spectrum associated with a single primary user (PU). The spectrum is sensed and the spectrum holes are detected to satisfy the requirement of the secondary user (SU).

**Key words:** Spectrum sensing, primary user (PU), Spectrum holes, Cognitive radio, Secondary users (SU), Deep Q-Network (DQN).

Received on DD MM YYYY, accepted on DD MM YYYY, published on -XO\



doi: 10.4108/_____

## 1. Introduction

In this era of technology and advancement the number of users adapting to wireless communication are increasing from day to day. But the bandwidth reserves to accommodate the increasing number of users is very limited. So, we need enhanced methods to provide the bandwidth to satisfy the requirements of the increasing number of users.

Spectrum sensing is a prime step in cognitive radio based dynamic spectrum management to gain the understanding of the radio environment. Regardless its influence in past generations, the research of spectrum sensing has attracted considerable interest from the wireless communications society. Spectrum sensing is a system initially created for military applications, to supply secure communications by spreading the signal over a large frequency band. Despite the fact that the fixed spectrum assignment outlook served well in the past, due to exceptional increasing in the number of users competing the spectrum has led to the deficiency of spectrum. Spectrum holes is a consequence of inefficient usage of spectrum. As a result, potential shift from fixed spectrum sensing to dynamic spectrum sensing is absolutely necessary to overcome the contemporary constraint. Dynamic Spectrum Sensing (DSS) is the crucial remedy for rational spectrum switching.

The fundamental scheme of a cognitive radio is to assist the secondary users to communicate over the

*Corresponding author. Email: author@emailaddress.com

spectrum allocated to the licensed users when they are not completely using it. Cognitive Radio performs on this dynamic Spectrum Management (DSM) principle which solves the issue of spectrum underutilization in wireless communication in an efficient manner.

Cooperative spectrum sensing (CSS) is utilized when the cognitive radio devices are scattered over various locations. It is conceivable for the cognitive radio devices to cooperate in order to obtain improved sensing accuracy than non-cooperative sensing does resulting in a feasible solution. In cooperative sensing, the cognitive devices share the sensing results with the fusion center for decision making. In this way the secondary users collaborate to perform reliable spectrum sensing.

We try to address this problem with the help of Deep Reinforcement based method called Deep Q-Network to perform the spectrum sensing operation.

## 2. Related work

The problem of wastage of spectrum has been widely studied in the literature. To direct the limitation of system dynamics the author in [1] used a method called improvident policy which can achieve better performance among different user frameworks based on the theory of partially observable Markov decision processes (POMDP).

The Deep Reinforcement Learning approaches are reviewed and studied in [3] to convey upcoming issues in networking and routing. The most important issue which is studied in this paper is the dynamic channel access. DRL is used when the states actions and state spaces are large data sets which cannot be solved using RL. DRL achieves optimal policy after training sets and produces best reward for an action.

To avoid collision of channels which occur when two or more users strive to transmit simultaneously over a common channel, or when two users attempt to transmit simultaneous in a half-duplex communication channel, the author at [2] proposed a Q learning based channel selection algorithm that is implemented in secondary users to determine the scanning order of channels and perform the cooperative spectrum sensing.

In recent years, the evolution of generations of wireless networks such as 5G. 5G is dynamic and flexible generation of mobile connectivity till date, leveraging cloud native applications and core. The upcoming spectrum sharing technologies in 5g network mainly use cognitive based sharing of the spectrum, they are carrier-based spectrum sensing, full duplex and spectrum database-based sensing. All these techniques are surveyed by the authors in [10] to develop the spectrum sharing methods in 5G and further. The problems which were in 2G, 3G and 4G were addressed and tried to overcome in 5G with the help of these methods of spectrum sharing. One of the most important features of 5G is the utilization of artificial intelligence for various tasks. Authors in [8] published a Deep Learning enabled spectrum sensing radio for the applications that use 5G such as social media application, live broadcasting and video calling as there require higher bandwidth and higher speed.

Systematic and well-structured learning algorithms are proposed in [6] by the authors to model as a spectrum access in CRN. The costs of secondary users are sustained as a result of the activity of primary users for a channel. To provide information to the secondary users the authors proposed a common channel which corresponds the primary user, using Radio Frequency spectrum in a given environment.

To solve the MDP problem authors in [11] published an actor-critic algorithm based to determine the optimal policy by updating optimization parameter problem. The value of all states are maximized simultaneously by optimal policy. If an optimal policy exists, then the policy that gives the maximum value is same for the present state and next state.

The most widely used cognitive radio patterns are discussed by the authors in [12]. The basic idea of CR has been implemented for the allowance of the harmony of various technologies and networks in the single shared spectrum. Further the authors addressed the measures to be taken to alleviate the deterioration of the performance because of interference.

The authors in [13] wrote about the importance of deep learning in any field of science. Deep learning uses neural networks to solve for huge data sets and train them to get maximum optimal policy that agent will follow. Deep learning has been a quantum leap in various technologies like image and video processing and speech processing. Needless to say, it is widely used in CR networks too.

Importance of Deep Learning for physical layer is discussed by the authors in [14]. End to End communication are reconstructed that looks to boost the performance of components in transmitter and receiver in a single process.

There has been climb in recognition of use of GPU in computers for general purpose applications. The authors in [15] studies about the importance of deep learning algorithms that are used in computers for GPU's. CNN (Convolutional Neural Networks) are approaches based on computer vision are studied in [15].

The authors in [16] reviewed the use of neural network in imaging. Machine Learning and Depp Learning has had a boost in in the industry related to imaging and inverse imaging. These algorithms can work with large data sets and provide optimal results easily.

Recently the increase in interest in NLP has made a way for the tasks which involve natural language processing. These techniques which are studied in [17] use a model for pre training and applies it to the object. Deep learning which involves natural language processing give state of the art results for the required method.

The authors in [18] studies the importance of big data ion emerging computational requirements for networking. The small base stations are used to manage huge data sets flowing. Due to this congestion happens and to avoid this big data plays a huge role.

AI enabled procedure is proposed in [19] which is used to predict the characteristics of the channel in MIMO indoor channels using convolutional neural networks. The CNN is used to establish the relation between transmitter and receiver locations.

Methods like data mining are implemented in big data for 5g in [21] to get the features of the model of the data traffic and probabilistic features to capture it. Indoor localization is a method which is popular and implemented by the authors in [21] to give the positioning of the users.

A deep neural network implemented with hessian matrix is studies in [23] .It gives the second order derivatives of the, matrix and uses gradient based optimization for approximation.

## 3. System model

The practical user scenario considered consists of two networks (Fig 1).

They are

a) Primary network
b) Cognitive network

The first network is the primary network which consists of the primary base station. This primary base station is associated with a particular bandwidth that it is intended to allocate it to the primary users based on their requirement. The primary users are the licensed users. The bandwidth that is associated with base station is the maximum bandwidth that can be allocated for the licensed users. Hence it is the licensed band. Each base station can consist of more than one licensed band.

The second network is the cognitive network which consists of all the secondary users. The cognitive network is a special network which is used to perform the advanced operations on the spectrum. various operations such as spectrum management, spectrum allocation and spectrum aggregation can be performed with the help of a cognitive radio. These secondary users sense the bandwidth associated with the base station.

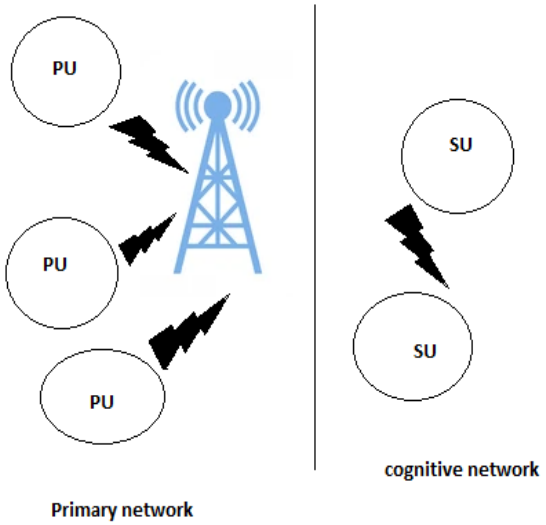


Fig 1 Illustration spectrum sensing

The spectrum sensing is usually performed on the primary network by the secondary network. The secondary user, usually present in secondary network, must sense the bandwidth and should utilize the vacant channels at a given time without interfering with the transmission of the primary user. If the associated channels are vacant then it should use the vacant channel to satisfy its requirement with the help of cognitive radio. If the channel turns out to be busy then it must not use the channel.

The spectrum hole is the time for which the given channel is vacant for a considered time period. The secondary user has to determine the spectrum holes. The concept of spectrum holes is illustrated in Fig 2. Effective determination of the spectrum holes in the real time is the main aim of the proposed solution. We try to predict the vacant spectrum holes with the help of DQN.

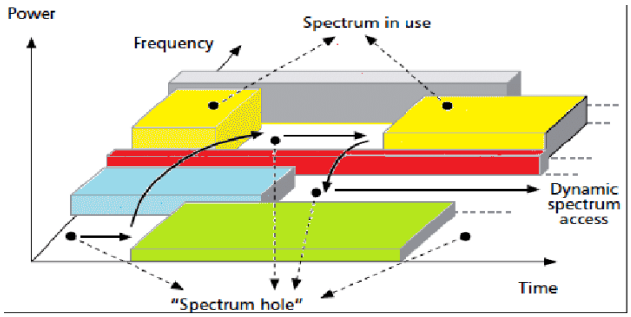


Fig 2 Illustration of Spectrum Holes

## 4. Methodology

### 4.1. Radio Frequency Environment and Frequency band for Spectrum Sensing

The environment that we consider to perform the spectrum sensing is radio frequency environment. In this environment we select the particular frequency band for sensing which is licensed. The primary users (PU) can access the frequency band that is allocated to them whenever they need. Hence the state of the channel, whether it is vacant or busy, is unpredictable depending on the nature and requirement of the PU's. So, this can be considered as a random process.

The SU's can be unlicensed user or a licenced user that is denied channel allocation. The channel allocation for a licensed user can be denied when all the channels associated with the base station are busy. So, the secondary users can access the other licenced channels using cognitive base station.

### 4.2. Spectrum Sensing by SU's

Each secondary user follows a deep reinforcement learning based algorithm. This DRL algorithm is used to predict the state of the channel. Each DRL method has

respective deep learning agent. A deep learning agent is an automatic or semi-automatic artificial intelligence (AI) - based system that uses deep learning to execute and enhance at its tasks. Each secondary user consists of a DRL Agent. This agent assess the environment and return the accurate state of the channel considering various parameters. The algorithm we are going to use in our proposed model is DQN.

### 4.3. Fusion center and Spectrum Decision

The main goal of the fusion center is to collect all the actions. The actions that are predicted by the Agent present in each of the secondary user is collected individually and then stored. It is a memory unit that consists of the action and the corresponding user that predicted the action.

The spectrum decision is taken based on the data that is collected by the fusion center. It follows a distributed process to find the best action suggested by the secondary users collectively. It just weighs the number of users that suggested the channel will be vacant to the number of users that declared that channel will be busy. Then it will select the action that is suggested by the majority of the users. If both the users suggested either cases are same, which is highly unlikely, then the decision will be taken randomly.

After spectrum decision, if the channel is available then the available vacant channel will be allocated to any of the SU using cognitive radio else then a different frequency band for sensing is selected in the radio frequency band and the whole process is repeated again.

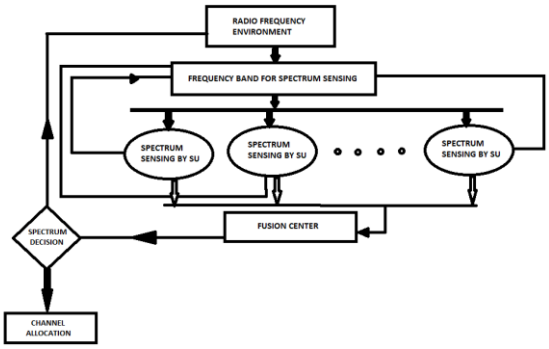


Fig 3 Block Diagram

## 5. Working of DQN

### 5.1. Deep Q- Network

The three most important aspects of DRL are states, actions and rewards. States (s) are a representation of the characteristics of current environment. Actions (a) are something a DRL agent can do to change these states based on the requirements of the user. And rewards $r_{(s,a)}$ are the utility the agent receives for performing the "right" actions. The main aim is to learn a "policy" which tells you which action to take from each state so as to try and maximize reward.

Each state in an environment is a result of an action performed on its previous state which in turn is a consequence of an action on its previous state. Nevertheless, storing all the information is very difficult and consumes a lot of memory. To counter this problem, we suppose that each state follows a Markov decision process, in which each state depends only in the previous state and transition from previous state to present state.

The number of states that are available depends upon the number of channels that the sensed spectrum consists. If the spectrum consists of 'N' channels then $2^N$ states are possible. Now, we assign a reward based on the transition of the states. We assume the reward obtained when the state associated with the required bandwidth demand as the central value. Now, for decrease in the number of vacant channels we reduce the demand by '1' and for increase in the number of channels we increment it by '1'.

Now, based on the rewards that are obtained now agent know exactly the actions that it must perform. It will perform the series of actions on the environment that results in the maximum total reward. This is called as the Q-value and this can be calculated with the following formula. [24]

$$Q(\acute{s},a) = r_{(s,a)} + \gamma \max[Q(s,a)] \quad \ldots. (1)$$

Here the $r_{(s,a)}$ corresponds to the reward obtained by performing an action (a) on state (s) resulting in state ś, $\gamma$ is the discount factor and Q(ś,a) denotes the Q-value of the previous state. This states that the Q- value obtained for a given state depends on the Q value and gamma. The contribution of rewards in the future is controlled by discount factor. It ranges from 0 to 1. This process is repeated until the acquired results are maximum.

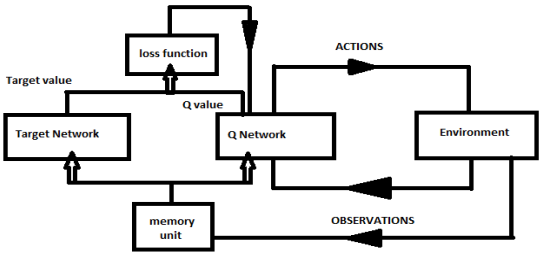


Fig 4 Block diagram of DDQN

### 5.2. Double Deep Q Networks

The double deep Q Networks (DDQN) is a variant of the DQN. This involves the usage of two neural networks to improve the stability of the overall system. The two networks are the Q-Network and the Target network. The DDQN is also included with a memory unit (shown in fig 8) which is used to perform the experience replay, a

process of memorizing and reusing the stored experience. To perform experience replay, we sample a mini batch from the stored experience of the agent. Instead of running Q function on state or action value the system stores the previous value and uses it to improve the next reward.

The function of this Q network is same as the DQN. The Q network generates the Q value and performs the action on the environment. If the same network is used for finding the predicted value and the target value, there would be huge difference between these two values. So instead of one neural network as in DQN two networks are used for learning in DDQN.

The target network generates the target value ($Y_i$) and finds the action that gives the maximum Q-value (policy). For each sample 'i' the target network tries to minimize the loss. The structure of Q – network and the target network are identical. At regular time intervals, the parameters for the target network are copied from the Q-network. The target value gets updated from one episode to the next episode. The policy and target value is calculated by using the following equation (2) and (3) respectively.[25]

$$A_{max} = \arg\max Q(S_i', A' | \theta_Q) \qquad \text{.....(2)}$$

The Q-value for the states ($S_i'$) after performing an action ($A'$) on state 's' for a sample 'i' are calculated and the action corresponding to maximum Q- value is obtained. This action is called as optimal policy and this process is called policy estimation. This is can be done with the help of Maximum argument (arg max) function. This helps in finding the optimal policy. The $\theta_Q$ represents the parameters with which the Q-network is initialized.

$$Y_i = R_i + \gamma \max A' Q'(S_i', A' | \theta_{Q'}) \qquad \text{....(3)}$$

In this equation (3) the $R_i$ is the target value ($Y_i$) of the previous state. The $Q'(S_i', A' | \theta_{Q'})$ in the target function is itself same as the Q-function. It is calculated for the state of an $i^{th}$ sample $S_i'$ which are obtained by performing an action $A'$ on state 's'. $\gamma$ is the discount factor. $\theta_{Q'}$ represents the parameters that are copied from the Q-network.

The loss function calculates the mean square value of Q - value and the target value for the $i^{th}$ sample in 'M' number of samples. This is called as loss. It is the variation of the target value from the q value and is used in the next training iteration. The loss is calculated by using the following formula. [25]

$$L = (1/M) \sum ( yi - Q ( Si , Ai | \theta Q ))2 \qquad \text{..... (4)}$$

## 5.3. Algorithm overview

i. Set the Q-network Q(s,a) with random parameter values $\theta_Q$ .
ii. Copy the same values to the target network.
$$\theta_{Q'} = \theta_Q$$
iii. For each training time step, for initial observation go to step (iv) else go to step (v).
iv. A random action 'A' is selected.
v. Choose the action for which the Q value function is maximum (policy).
$$A = \arg A \max Q(S,A| \theta_Q)$$
vi. Execute action A.
vii. Get the reward R and next observation S'.
viii. Store the experience (S,A,R,S') in the experience buffer.
ix. Sample a random mini-batch of M experiences (Si,Ai,Ri,S'i) from the experience buffer.
x. If S'i is a terminal state then go to step (xi) else go to step (xii)
xi. Set the target value function yi to Ri.
xii.
$$A_{max} = \arg A' \max Q(S_i', A' | \theta_Q)$$
$$Y_i = R_i + \gamma \max A' Q'(S_i', A' | \theta_{Q'})$$
xiii. Across all the sampled experiences, replace the target parameters by loss L.
$$L = (1/M) \sum ( yi - Q ( Si , Ai | {}^{\theta}Q ))^2$$

The Q network is designed with some random parameter values $\theta_Q$ (step i) and these same values will be copied into the target network (step ii). For the initial step, a random action is selected (step iv) and executed (step vi). After executing the action the rewards and the next observation are obtained. These are stored in the experience buffer (step viii). A random mini batch is considered from the stored values from the experience buffer (step ix). This mini batch is used by the target network for policy estimation by using equation (2) and for the calculation target value using equation (3) (step xii).Then the loss is calculated using equation (4) (step xiii).In the next iteration the calculated target value will be updated to reduce the loss and the policy obtained is considered as new action (step v).

If the terminal conditions are not met then the previous action for which the Q-value is maximum is considered and the process will be repeated.

If the terminal conditions are met then the value of Yi is set to Ri and the iterations will be stopped automatically (step xi). In this way the above algorithm works to perform the spectrum sensing using DQN agents.

# 6. Simulation

## 6.1 Cognitive radio network

To mimic the primary user incumbent is utilized. Incumbent refers to the current holder of a position. The operation of the secondary user can be realized by using CR_CPE (Customer premises equipment).It is a service provider equipment that is located on the customer's premises. We consider an incumbent and 4 cr cpe which are connected to a single base station as a general scenario.

The incumbent is associated with a base station. The operating frequency of the incumbent is asserted in the

base station. Operating frequency is the frequency band at which the incumbent operates. The incumbent can use the operating frequency band whenever it needs. This band can range from 54 MHz to 862 MHz, the bandwidth of every channel is 6 MHz

For example, if the operating frequency is set from 54 MHz to 72 MHz,

• Channel 1 will be 54 to 60 MHz
• Channel 2 will be 60 to 66 MHz
• Channel 3 will be 66 to 72 MHz

To control the operation of the incumbent and represent it as primary user ON and OFF duration are specified. ON duration is the duration of time for which the primary user i.e., incumbent user operates. OFF duration is the time interval between two successive ON durations of an incumbent.

The channel switching occurs whenever the state of a channel is changed, busy to vacant or vacant to busy. Channel switching is a process where the secondary user switches the channel when the primary user of the channel comes back to use it. This occurs because the secondary user continuously senses the frequency band allotted and tries to utilize it whenever it is vacant. So, for continuous utilization of channel, if needed, the secondary user must be assigned channel continuously whenever the current channel becomes busy. To make this possible the channel switching is needed.

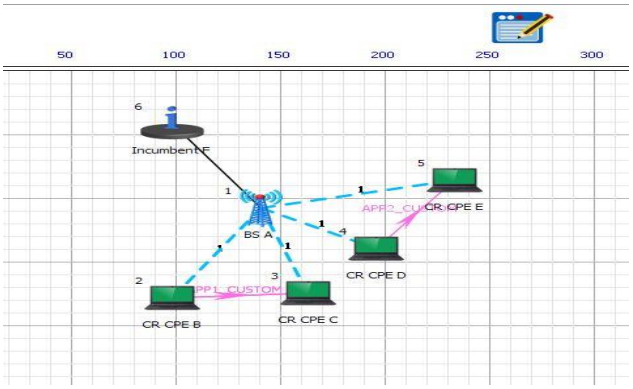


Fig 5 Environment

## 6.2 Deep Q-Networks

Initially a random state transition probabilities are assumed in the created MDP environment to mimic the usage of channels by a hypothetical primary user. A total of 3 channels which results in 8 states are considered. The rewards associated with each state transmission is assumed. The information regarding the states and the actions are obtained from the defined environment. This information is used by the two different networks. So we designed a target network Fig 9 using different layer functions.

The target network is created by using mainly three layers. They are image input layer, relu layer and fully connected layer. The image input layer is used to take the inputs in the form of the row vectors. The relu stands for rectified linear unit, this layer performs the threshold operation and performs the relay operation from one layer to another. In a fully connected layer the bias vector is added to the weight matrix after multiplying it by input.

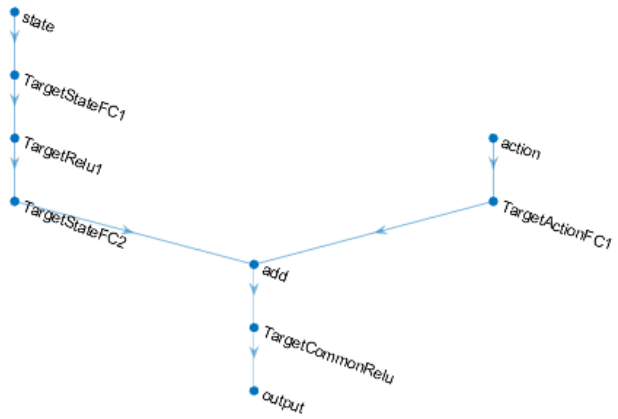


Fig 6 Target Network

The target network consists of 2 parts, the state and action paths (Fig 9). The state path is used to get the states and the action path is used to get the action from the environment. These two paths are combined to form a common path through which the target network can access the both states and actions of the current environment.

After the target network is created the agent is created. Every time when the agent starts sensing the environment it resets all the states and actions but continues to store the experience. Then it performs the sequence of actions to maximize the reward and the action which gives the maximum corresponding reward is considered to be the optimal policy. The policy is the agent's behaviour at a given instant of time. To create an agent we specify some constraints on the basis of which the agent works. These are called as the agent options.

The table 1 shows the agent options that are used in creating an agent. The agent options that we specify plays an important role in the working of the agent. The operation of the agent can be regulated by varying these values.

| S.No | Agent options | value |
|---|---|---|
| 1 | Use double DQN | true |
| 2 | Target update method | periodic |
| 3 | Target update frequency | 4 |
| 4 | Experience buffer length | 1000000 |
| 5 | Discount factor | 0.9999 |
| 6 | Mini batch size | 256 |

Table 1: Agent Options

These options are explained below

1. Use Double DQN: It is a method which uses two similar neural networks. First one learns during the experience replay, in the same way DQN learns, and the second network one copies the parameters of the first network's last episode. To specify the use DDQN

the value must be set to 'true' and the 'false' specifies that DQN is used

2. Target Update Methods: The target parameters are updated by DQN agents using the periodic method. This method updates the target parameters periodically without using smoothing. To define the update period, use the Target Update Frequency parameter.
3. Target Update Frequency: It is defined as the number of steps between each target critic updates, it is a positive integer.
4. Experience Buffer Length: Experience buffer size, is a positive integer, which is updated when the agent computes using a mini-batch of experiences randomly sampled from the buffer during the training period.
5. Discount factor: Discount factor applied to future rewards during training, it tells how important the reward is. It is a value between 0 and 1.
6. Mini Batch Size: It is a size of random experience in a mini-batch, given as a positive integer. In each training episode, the agent samples the experiences randomly from the experience buffer when. Large mini-batches reduce the variance while computing the gradients but increase the computational complexities.

Once the agent is created then it is trained within in the designed environment. The agent performs the main training operation by following the algorithm discussed and within the values of the training options that are specified in table. Training is the process in which the agent improves its accuracy by performing the actions gaining experience and using in gained experience to determine the next action. The table 2 mentions the training options that are used to train the agent.

| S.No | Training options | value |
|---|---|---|
| 1 | Max Episodes | 500 |
| 2 | Max steps per episode | 100 |
| 3 | verbose | true |
| 4 | Plots | Training-progress |
| 5 | Stop training criteria | Average Reward |
| 6 | Stop training value | 15 |

Table 2: Training Options

These parameters are explained below
Episode: It is the simulation length for which the system ends in a terminal state or all the intermediate states that occurs between initial state and the final state.

1. Maximum number of episodes: To train the agent Maximum number of episodes is specified as a positive integer. Regardless of alternate criteria for termination, training terminates after Max Episodes.
2. Maximum steps per episode: Maximum number of steps to run per episode is specified as a positive integer.
3. Verbose: It display training result on command line which are specified as logical values false (0) or true (1).
4. Plots: It is used to display progress of the training agent with Episode Manager.
5. Stop training criteria: It represents the terminating condition of the training. The average reward is considered as the stop training criteria. The training will be terminated when the average training reward equals or exceeds the stop training value.
6. Stop Training value: It is a scalar that specifies the critical value for which the training of the agent will be terminated.

With these parameters the agent will undergo training and perform the sensing operation.

## 7. Results

By performing the simulation by taking the above considerations into account the following results are obtained.

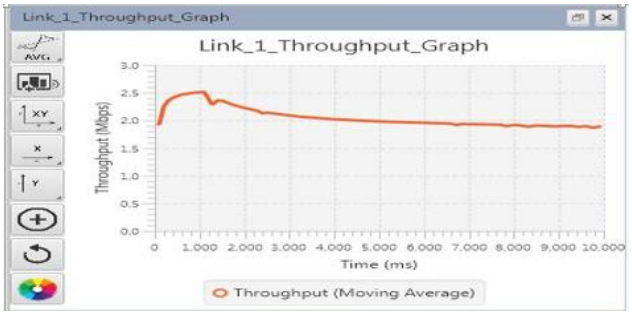


Fig 6 Link 1 throughput graph

The fig 6 shows the throughput plot of link 1 with respect to time. Link one corresponds to the connection between incumbent and base station i.e., primary user and the base station. The rate at which data packets are successfully transmitted in unit time in the network is called Throughput. So through the activity of the primary user can be analysed.

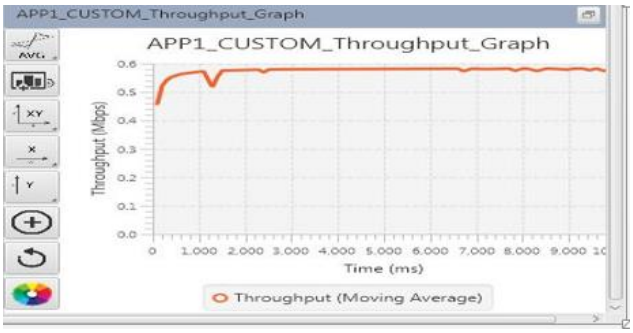


Fig 7 App1_custom Throughput graph

This plot (fig 7) shows the throughput of App1_custom link. This corresponds to the connection between CR_CPE B and CR_CPE C i.e., this is the

connection between the two secondary users which are sensing the channel and utilize it whenever available. The throughput is varied as follows initially it is less then it increased gradually and became stable after a sudden surge.

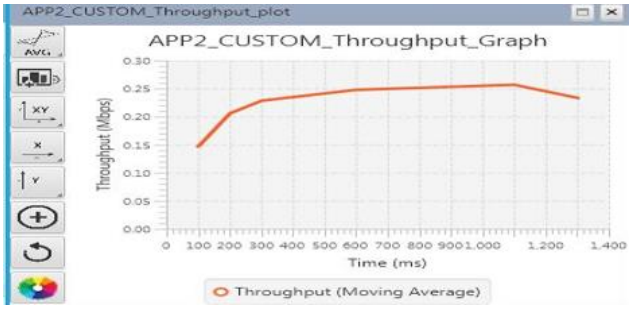


Fig 8 App2_custom_throughput graph

This plot (fig 8) shows the throughput of App2_custom link. This corresponds to the connection between CR_CPE D and CR_CPE E. Similar to the previous case. The throughput is varied as follows initially it is less then it increased gradually and became stable sustained for a period and then declined gradually. From the variation between the fig 8 and fig 7 it is evident that the throughput variation is different. This happens because even when the vacant channel can be utilized by only one user to avoid the interference and loss of information.

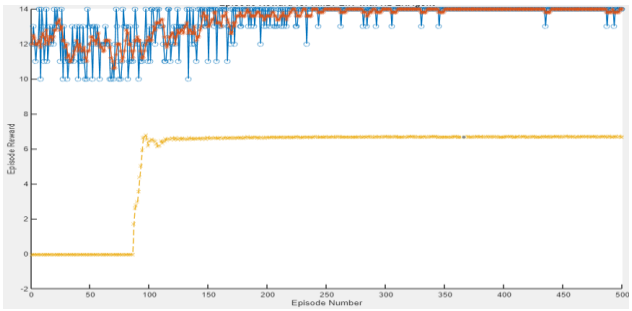


Fig 9 Training of DQN agent

This graph (fig 9) gives the training of the DQN agent. The red curve indicates the average reward, blue curve indicates the reward for each episode and the yellow curve indicates the target value. The average reward is calculated by taking the average of the episode reward (blue curve) of all the samples until the considered sample. Since it is the DQN agent it uses only one neural network hence the Target value (yellow curve) is very different from the Average reward obtained (red curve). From fig 9 it is evident that the distinction between the target value and the average reward is very. This results in the higher loss. Hence the stability is less.

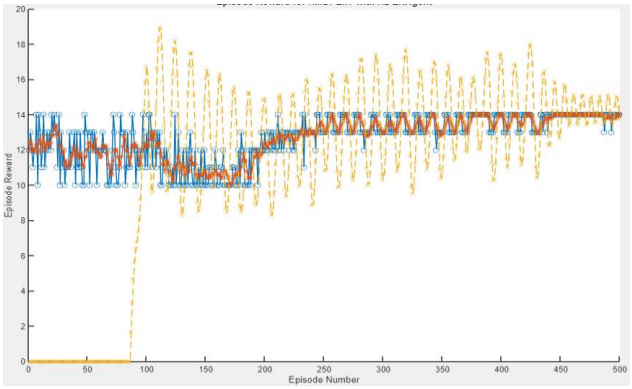


Fig 11 training of DDQN agent

This graph (fig 11) shows the training of a DDQN agent. This agent uses two interdependent neural networks to calculate the Q-value and the target value respectively. From the plot it is evident that the distinction between the target value (yellow curve) and the Average reward (red curve) is very less. Hence the loss obtained will be less. So, the stability will be more compared to the DQN agent.

The target value and the average reward for both the agents are shown in table 3. The maximum obtainable reward is 14. So, it is same in both the networks but the average reward must be closer to the target value. Therefore from the table 3 it can be inferred that DDQN agent is more stable for spectrum sensing compared to DQN agent.

| | **Final episode Target value** | **Average Reward** |
|---|---|---|
| **DQN** | 6.6701 | 14 |
| **DDQN** | 13.7329 | 14 |

Table 3: Comparison of DQN with DDQN

## 8. Advantages and Applications

Hidden node problem is significantly reduced by using a cooperative spectrum sensing system since more receivers can build a precise picture of the spectrum usage in the network. Agility increases because number of secondary users are increased. The sensing is more accurate because of the cooperation between the users.

In cooperative sensing, the nodes performing the spectrum sensing results in more accurate channel signal detection. Hence the number of false alarms are reduced. With the increase in the development of the cellular technology the generations are getting updated from day to day adopting new and advanced methods.

One of the recent update in cellular technology is the proposal to introduce Artificial intelligence (A.I) to further improve the performance of the cellular network. So, using machine learning for efficient spectrum sensing can become an integral part of the future generations.

## 9. Conclusion

Wireless communication is one of the most important aspects of the present world. But as the number of users increasing day by day it is becoming very difficult to accommodate all the users. One of the constraint that restricts the number of users is available bandwidth. So, we have to device new methods for the effective utilization of bandwidth and provide services to all the users.

The main aim of the spectrum sensing is to locate the spectrum holes effectively. To achieve this objective we employ Deep Q-Network and its stability is improved by using the Double Deep Q-Network. This is a Deep reinforcement learning method. Using deep learning enables us to handle large state action space. The usage of Reinforcement learning along with deep learning enables us to determine the spectrum holes with optimal memory requirement.

## References


[1] Yun Zeng Li, We sheng Zhang published "Deep Reinforcement Learning for Dynamic Spectrum Sensing and Aggregation in Multi-Channel Wireless Networks"

[2] Wenli Ming, Xiao yen Huang et al [2] published "Reinforcement Learning Enabled Cooperative Spectrum Sensing in Cognitive Radio Networks"

[3] N. C. Luong, N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," 2018, http://arxiv.org/abs/1810.07862.

[4] V. R. Konda and J. N. Tzatzikis, "Actor-critic algorithms," in Advances in Neural Information Processing Systems, S. A. Silla, T. K. Leen, and K.-R. Muller, Eds., Vol. 12, MIT Press, Cambridge, MA, USA, 2000.

[5] R. S. Sutton and A. G. Barton, Reinforcement Learning: An Introduction, MIT Press, Cambridge,MA, USA, 1998.

[6] Z. Han, R. Zheng, and H. V. Poor, "Repeated auctions with Bayesian nonparametric learning for spectrum access in cognitive radio networks," IEEE Transactions on Wireless Communications, vol. 10, no. 3, pp. 890–900, 2011.

[7] A. Ali and W. Hamada, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," IEEE Common. Survey. Tutorials, vol. 19, no. 2, pp. 1277–1304, 2017.

[8] Munib Aalam Khan, Amir Zeb Shaikh et al [8] published "Deep Learning Enabled Spectrum Sensing Radio for Opportunistic Usage

[9] R. Tandra and A. Sanai, "SNR walls for signal detection," IEEE J. Sel. Top. Signal Process., vol. 2, no. 1, pp. 4–17, Feb. 2008.

[10] F. Hu, B. Chen, and K. Zhu, "Full spectrum sharing in cognitive radio networks toward 5G: A survey," IEEE Access, vol. 6, no. c, pp. 15 754– 15 776, 2018.

[11] Fan Wu, and Supeng Leng et al [11] published "Reinforcement Learning based Cooperative spectrum sensing

[12] S.K. Sharma, T.E. Beagle, S. Chatzinotas, B. Otters ten, L.B. Le, and X. Wang, "Cognitive radio techniques under practical imperfections: A survey," IEEE commune. Survey. Tutorials, vol. 17, no. 4, pp. 1858–1884, 2015.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[14] T. O'Shea and J. Hyoids, "An introduction to deep learning for the physical layer," IEEE Trans. Cong. commune. Netw., vol. 3, no. 4, pp. 563–575, Dec. 2017.

[15] E. Nishan and B. C, ic, o, "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation," in Proc. IEEE Mediterranean Conf. Embedded Computer. (MECO), Bar, Montenegro, Jun. 2017, pp. 1–4.

[16] A. Lucas, M. Iliad's, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," IEEE Signal Process. Mag., vol. 35, no. 1, pp. 20–36, Jan. 2018.

[17] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," IEEE Computer. Intel. Mag., vol. 13, no. 3, pp. 55–75, Aug. 2018.18

[18] Y. Wu, F. Hu, G. Min, and A. Zomaya, Big data and computational intelligence in networking, Florida: Taylor & Francis, 2017.

[19] L. Bai, C.-X. Wang, J. Huang, Q. Xu, Y. Yang, G. Goussetis, J. Sun, and W. Zhang, "Predicting wireless mm Wave massive MIMO channel characteristics using machine learning algorithms," Wirel. Commun.Mob. Computer., vol. 2018, Aug. 2018.

[20] C.-K. Wen, W.-T. Shih, and S. Jinn, "Deep learning for massive MIMO CSI feedback," IEEE Wireless Commun. Lett., vol. 7, no. 5, pp. 748–751, Oct. 2018.

[21] J. Huang, C.-X. Wang, L. Bai, J. Sun, Y. Yang, J. Li, O. Tirkkonen, and M. Zhou, "A big data enabled channel model for 5G wireless communication systems," IEEE Trans. Big Data, vol. 6, no. 5, Mar. 2020.

[22] Z. Zhang, H. Chen, M. Hua, C. Li, Y. Huang and L Yang, "Double coded caching in ultradense networks: Caching and multicast scheduling via deep reinforcement learning", IEEE Trans. commune. (Early Access), Nov. 2019.

[23] L. Zhang, W. Zhang, Y. Li, J. Sun, and C.-X. Wang, "Standard condition number of hessian matrix for neural networks," in Proc. IEEE Int. Conf. commune. (ICC), Shanghai, China, May 2019, pp. L.

[24] Book : "Machine Learning" by Tom M. Mitchell chapter 13

[25] Ramkumar Raghu1, Pratheek Upadhyaya1, Mahadesh Panju1, Vaneet Agarwal1,2, and Vinod Sharma1."Deep Reinforcement Learning Based Power control for Wireless Multicast Systems".In: 57th Annual Alleration Conference on Communication, control and Computing (Allerton) Oct,2019.