

Transmission Power Control Based on Cross Layer Routing Optimization Technique in Cognitive Radio Network

¹S. Aravindkumar, ²D. Saraswady, ³S. Sedhumadhavan

Submitted: 27/04/2023

Revised: 28/06/2023

Accepted: 07/07/2023

Abstract: Cognitive radio networks have been proposed as a feasible option for the fifth generation (5G) wireless system to address the different demands. These networks utilise intelligence to access a principal user's underutilised channel. Cognitive radio networks have developed as a potential solution to the problem that permits unlicensed users to get dynamic spectrum while licenced users remain inactive. One of the many critical cognitive radio processes is channel assignment to the unlicensed user. Due to the variability of channel propagation characteristics, sporadic availability of licenced channel, frequent hand-offs, and demand for critical user security, finding a viable route is more challenging. Additionally, the capability of spectrum management at all network levels is required by the inclusion of opportunistic spectrum access in a cognitive radio network. If there are too many levels, management costs increase. As additional layers are added, performance becomes slower. In cognitive radio networks, this study proposes a unique method for cross-layer model-based power transmission management with routing optimisation. Here, the Levenshtein cross layer model is used to manage power transmission using a software-defined spectrum. Cross layer-enabled transceiver takes place in two separate lower layers; physical (PHY) and data link layer (DLL). Then, reinforced multilayer Q-graph colony optimisation is used to do the routing optimisation. Throughput, lifespan, jamming prediction, energy efficiency, routing delay, and packet delivery ratio are all included in the simulation study. Proposed technique attained throughput of 96%, lifetime of 73%, jamming prediction of 82%, energy efficiency of 65%, routing latency of 55%, packet delivery ratio of 88%; existing LEACH attained throughput of 92%, lifetime of 68%, jamming prediction of 77%, energy efficiency of 59%, routing latency of 52%, PDR of 85%, CWSN attained throughput of 95%, lifetime of 72%, jamming prediction of 79%, energy efficiency of 63%, routing latency of 53%, packet delivery ratio of 86%. The performance of proposed CLM-CRN-MLT model increases the efficiency of the network and attains power consumption.

Keywords: cross layer model, power transmission control, routing optimization, machine learning, cognitive radio networks

1. Introduction:

Cognitive Radio Wireless Sensor Networks (CRWSN) utilise battery-powered nodes. Lack of energy is a serious issue with CR-WSN, especially in situations like wartime where quick and forceful response is required. The performance of CR-WSN is hampered by node battery level. Energy consumption to send a packet is a major challenge that researchers must overcome in order to build a routing protocol for CR-WSN. In CR-WSN, a sizable number of nodes are present. Each node in the CR-WSN is limited by the battery. The foundational technology for next-generation wireless networks is cognitive radio. Each node in a cognitive radio network (CRN) has a cognitive radio for wireless communications that makes use of modern advancements in RF model, signal processing, and communications software [1]. Because such a node may

access spectrum dynamically, a CRN has a lot of potential to increase spectrum efficiency. Further well-known primary/secondary network arrangement, cognitive radio is a potential for many significant applications due to its capacity to sense, adapt, and learn. For the US military, public safety, and future mobile base stations [2], CR is, for example, the most crucial technology for radio interoperability. A CRN has some distinctive characteristics. There may be a variety of available frequency bands at a node, each of varying sizes. A node can use multiple available frequency bands simultaneously because CR is software-based. From the perspective of wireless networking, these new CRN features present an entirely new set of algorithm design and protocol implementation research issues. As a result, use of CR method in distributed scenarios is still in its infancy, and several unsolved research issues are discussed in [3]. In order to particularly address problems of end-to-end CR performance over multiple hops as well as challenge of shielding PU transmissions from interference with limited environmental knowledge, a CR routing protocol (CRP) for ad hoc networks is suggested in this research. Traditional routing algorithms try to optimise latency and hop count from start to finish

¹Department of Electronics and communication engineering, Puducherry technological university, Puducherry-605014, India.

²Department of Electronics and communication engineering, Puducherry technological university, Puducherry-605014 India.

³ Department of Electronics and communication engineering ,Rajiv Gandhi College of engineering and technology, Puducherry-605014

Corresponding Author : S.Aravindkumar

for wireless ad hoc networks. Traditional routing techniques either greedily route packets depending on target location or use limited network data spanning many hops to optimise choice of path [4]. Additionally, they employ network-wide broadcasting without localisation data. The literature on these protocols is extensive. These methods, however, are not appropriate for CR operation since they do not permit simultaneously choosing the spectrum band or taking into account the impact that routes may have on other licenced devices using spectrum. Several works on CR networks have recently been proposed to address these issues [5, 6].

The paper is organized as follows: Section II lists the published work in relevant areas. Section III describes proposed design and methodology. Section IV explains simulation of several scenarios and provides extensive experimentation to investigate our approach. Sections V discuss flow of proposed algorithm, testing in field to give implication of our approach and conclusion.

2. Related works:

Studies have indicated that the clustering technique introduced by the CWSN's routing protocols can enhance network performance. LEACH, a common cluster routing protocol, is presented to CWSN. In this paper, LEACH and TEEN protocols were compared, as well as TEEN and the advanced threshold sensitive energy efficient sensor network (A-TEEN) protocol [7]. Work [8] suggested that the Internet needs a knowledge plane separate from data as well as control planes. According to author [9], constructing such a "cognitive network" would necessitate AI-based cognitive methods rather than incremental method ones. Intelligence must be incorporated into CRN architecture as well as protocols across stack in order to assist CRNs in becoming cognitive networks. Some obstacles that learning algorithms in CRNs face, as outlined in [10]. Numerous sensing methods have been proposed during the past ten years [11] and are based on matching filter, energy detection, cyclostationary detection, wavelet detection, and covariance detection [12]. Cooperative spectrum sensing was also suggested in [13] as a way to address hidden terminal issues that are inherent in wireless networks and improve sensing accuracy. Cooperative CR

have also been looked at in the literature in recent years, as in [14]. [15] contains recent surveys on cognitive radios. To be genuinely cognitive, CR must be able to learn as well as reason in addition to being aware of its environment [16]. Following the trailblazing concept of [18], a cognitive engine, which was recognised as foundation of a CR [17], can offer these characteristics. Using machine learning algorithms, a cognitive engine coordinates the cognitive radio's actions. However, the application of machine learning algorithms to cognitive radios has only recently received more attention [19]. These methods are categorized as either supervised or unsupervised learning. For cognitive radio applications, the authors of [20] have thought about using support vector machines and supervised learning based on NN. For DSS applications, unsupervised learning, such as reinforcement learning (RL), is considered in [21]. In [22], it was demonstrated that the distributed Q-learning algorithm works well for a specific cognitive radio application.

3. System model:

A multi-hop CRN comprising M PUs and N SUs is considered. The fixed spectrum allocation regulation dictates which spectrum bands are allotted to PUs. When SUs discover that the PUs do not own any frequency bands, they opportunistically transmit data even though they do not occupy any licensed channels. Data Transmission Channel (DTC) as well as Common Control Channel (CCC) are two types of channels that are accessible from each SU node n_i . Data transmission relies on the DTC, which can be represented as $C_i = c_1, c_2, \dots, \text{for } SU \ n_i \text{DTC. } c_m$; SUs, on the other hand, use the CCC to exchange the negotiation. If at least one common $DTC_c C_i C_j$ exists, a directed communication link is established at any time between SU n_i and n_j . Figure 1 depicts the model of the network. The two centralized Primary User (PU) networks and the multi-hop CRN coexist in the networking scenarios. Data packets are generated by the source SU and sent to destination node via intermediate SUs in a multi-hop fashion, as depicted in Figure 1. Each PU uses a licenced frequency to interact with the PU base station, while intermediate SUs in every hop transmit data via PU channel when spectrum band is not in use.

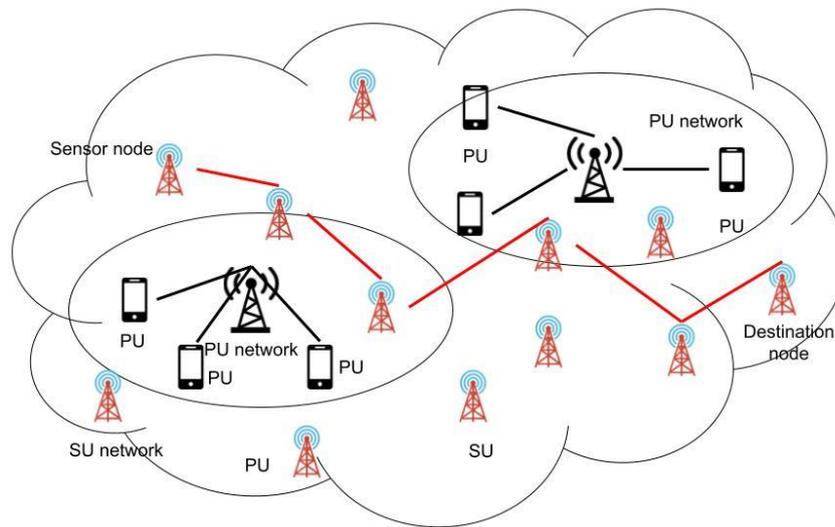


Fig-1. Multi-hop cognitive networking scenarios

Additionally, each SU node has a specific queue for each traffic flow, and each data stream's packet arrival procedure runs independently of the others. The occupation paradigm for PUs is thought of as an ON/OFF process. Equation (1) depicts probability density function (PDF) of OFF periods, during which PUs do not occupy channel.

$$f(t) = \begin{cases} \theta_d e^{-\theta_d t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (1)$$

where θ_d denotes the PU's departure rate and $f(t)$ is PU channel's idle probability at time step t . As a result, eq. (2) is used to represent the likelihood that the PU channel's idle period will be larger than its duration τ :

$$P(t \geq \tau) = \int_{\tau}^{\infty} f(t) dt = e^{-\theta_d \tau} \quad (2)$$

where τ is the amount of time the PU channel has been idle. The likelihood of PU and SU colliding in the duration is then given by equation (3):

$$P_{\text{collision}} = 1 - P(t \geq \tau) = 1 - e^{-\theta_d \tau} \quad (3)$$

to get the predicted mean μ and variation σ for the PU departure rate, which is expressed as θ_d (μ , σ). The parameterized spectrum statistic is believed to be nearly static in this work because of how slowly it changes. Only a small amount of positioning technology allows each SU to determine its own location.

Spectrum based energy efficient Levenshtein cross layer model:

It is envisaged that centralised PU networks will coexist with a distributed CRN that has N stationary SUs (SU1, SU2,..., SUN). The transmitter and receiver of nodes A

and B must be set up to use the same channel in order for them to communicate with one another. When employing dedicated radio interfaces to communicate with previous-hop and next-hop, channel switching every packet is not required. SU j 's interface 1 is set to channel ch_2 , SU i 's interface 2 is set to channel ch_1 , and all nodes' interface 0 is set to CCC for communication with SU k . Node j no longer needs to change the channel at its interfaces in order to accept, forward, and broadcast packets. The interface must switch from the configured channel to recently chosen channel if PU shows up on ch_1 or ch_2 again. In our suggested research, this channel change is conducted out locally between impacted nearby nodes without disrupting rest of route.

Cross-layer model

The proposed study takes into account cross-layering across the lowest three protocol stack levels, which are Network, Link, and Physical (see Fig. 2). For CRN, cross-layering is essential. Spectrum sensing, data or control packet transmission and reception, and other physical layer processes are important as well. The IEEE 802.22 standard has imposed limitations on sensing performance rather than calling for a particular approach. The limitation is to set the detection probability P_d at 0.9 or false alarm rate P_f at 0.1 to safeguard PU from SU gearbox. Needed minimum sensing time for each channel as well as total number of channels to be sensed determine sensing length. Sensing periods per channel vary between different sensing techniques.

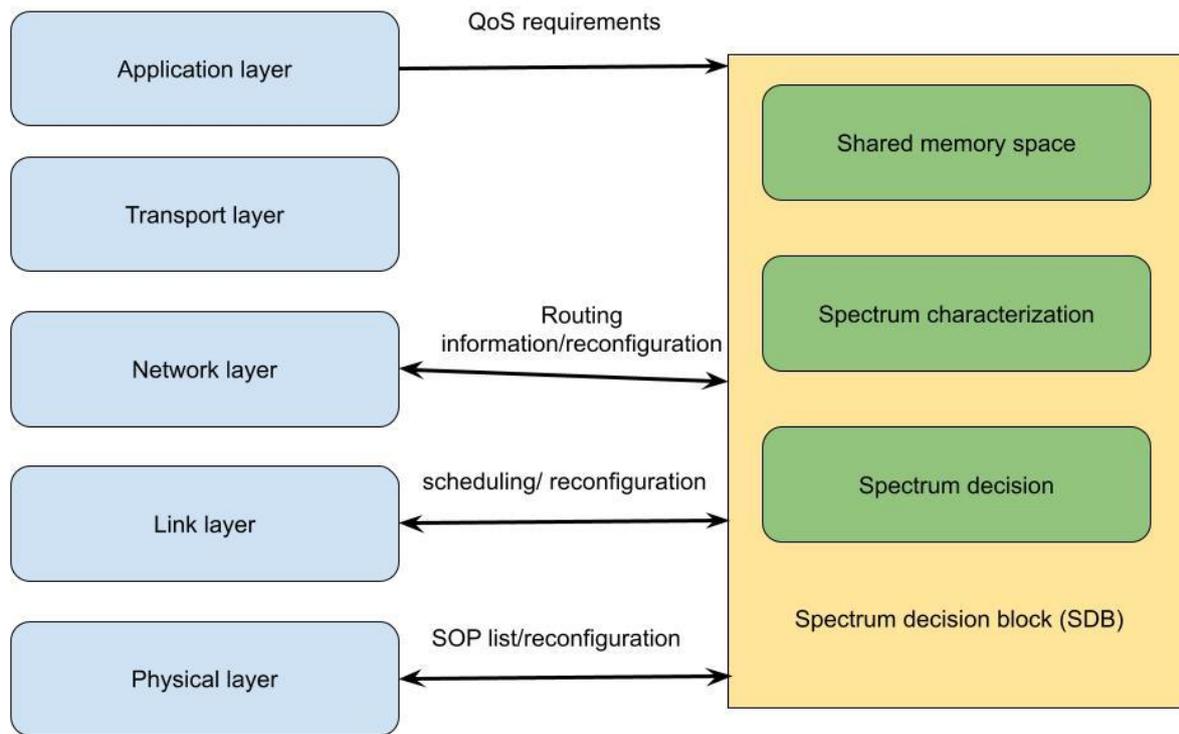


Fig 2. Cross-layer model

Due to the lower computational and implementation needs of Energy Detection (ED), cooperative sensing is assumed. The ED approach compares signal power to a threshold to assess whether transmission is occurring on the sensing channel. As a result, SUs are configured to pause transmission during sensing period to prevent false alerts [9]. SUs consequently switch between sensing and transmitting. The physical layer characteristics must be changed in line with selected transmission spectrum because these values differ for various wireless channels. As a result, SDB provides input to the physical layer in form of reconfiguration parameters. The network layer offers the channel list of one-hop neighbours. For the topology of the CRN, the existence of shared channels between the nodes is just as crucial as their physical separation. Then, SDB runs the channel choice calculation to select the reasonable channel at each bounce after receiving input from three layers. Chosen channel is sent into network layer during route construction as well as used at link layer to choose parameters for physical layer's reconfiguration. Wireless devices that are spectrum aggressive (CRs) are designed to operate over a broad frequency range. As a signal moves across a wireless channel, its strength diminishes with increasing distance. Theoretical maximum bit-per-second capacity of the link between SU i and its neighbour SU j utilising channel l is given by

$C_l(i, j) = B_l \times \log \left(1 + \frac{P_r(i, j)}{B_l \times N_0} \right)$ where $P_r(i, j)$ is received power at SU j located at (i, j) from SU i . N_0 is thermal

noise power density, and B_l is the channel l 's bandwidth. Equation (4) provides the non-Line of Sight (LOS) path between SU i and j as described by $P_r(i, j)$ as follows:

$$P_r(i, j) = P_f(i, j) \left[a(i, j) \left(\frac{d(i, j)}{d_{ref}} \right)^{-n} \right] \eta_s \quad (4)$$

According to equation (5-7), the amount of time (T_{pkt}) needed to send a packet of length S bytes over a connection (i, j) depends on both link's capacity and distance between the nodes.

$$T_{pkt}(i, j) = T_{tr, l}(i, j) + T_{pr, l}(i, j) \quad (5)$$

$$T_{tr, l}(i, j) = \frac{S \times 8}{C_l(i, j)} \quad (6)$$

$$T_{pr, j}(i, j) = \frac{d(i, j)}{v} \quad (7)$$

$T_{tr}(i, j)$ is amount of time needed to transmit a single packet over link (i, j) , $T_{pr, l}(i, j)$ is amount of time needed to propagate packet's initial bit over link (i, j) , and v is propagation velocity, which is 3×10^8 m/s. The propagation distance in ad hoc networks is typically less than 1 km, making $T_{pr}(i, j)$ very modest compared to $T_{tr, l}(i, j)$, and it is frequently disregarded. Theoretically, n data packets can be sent in nT_{tr} seconds, but only if the channel is open for the entire transmission. But in case of CRNs, things are different since SUs transmit on PU channel, and length of time PU channel is open is determined by the PU activity pattern.

The Levenshtein nearest centroid classification (LNCC) model, which is used in proposed study, uses

Levenshtein distance calculation to represent user signal's attributes before feeding it into NC classifier for classification. In this study, energy for forecasting users on spectrum band serves as starting point for LNCC model. As shown in eq. (8), energy received at i th CR user in k th sensing area is thus written:

$$S_{ik} = \frac{\sum_{j=1}^N E_{ik}(j)}{n} \quad (8)$$

Energy sample received by k th sensing region for associated i th CR user is represented by $E_{ik}(j)$ in equation above, where N is total number of samples. This study uses NC, an LNCC, to classify observations using quantitative variables using the energy detection function mentioned above that is given in Equation (9). $Dis(a_i, C)$ should represent the distance between the centroids represented by $C = C_1, C_2, \dots, C_n$. Each centroid representing either H_0 or H_1 is measured in terms of distance. The present sensing result is classed as either H_0 or H_1 based on the measured distance. Levenshtein distance is used to calculate distance first, followed by the NC classifier for classification. The exact distance between vectors is computed using the Levenshtein distance. Due of the timeframe, each collaborative virtual sensing method used is unique.

Therefore, the Levenshtein distance, which examines the distance between current sensing result and sensing classes, is used in this work to evaluate the similarity between sensing outcomes. The mathematical expression of Levenshtein distance between the sensing results u and v is provided in eq. (9):

$$Lev_{ww}(p||q) = \begin{cases} \max(p, q) & \text{if } \min(p, q) = 0 \\ \min(Lev_{mp}(p-1, q) + 1, \\ \min(Lev_{mw}(p, q-1) + 1, \\ \min(Lev_{ww}(p-1, q-1) + 1_{wp;rq}) \end{cases} \quad (9)$$

According to Equation (9), 1_{upvq} is the function that equals 0 when $up = vq$ and 1 otherwise. $Lev_{uv}(p||q)$ on the other hand, is the measurement of the separation

between first p sensing regions of u and the first q sensing regions of v . As previously stated, the results of sensing have p elements that correspond to u and q elements that correspond to v . The Levenshtein distance function is used to compare the sensing reports from the training and test samples. If the distance is smaller, the output of the classifier is robust. Otherwise, the result is not strong. If centroids were taken into account when calculating the a_p , for example, the sensing report a_p would have to fall under this category. The candidate set of centroids for a_p is denoted as $N(a_p)$ such that $N(a_p) \in C_1, C_2, \dots, C_n$. The $N(a_p)$ is used to evaluate the present sensing outcome in conjunction with every member of each sensing class. As $M(a_p)$, the membership function is abbreviated. Additionally, define EV_l as circumstance in which w elements in $M(a_p)$ exceed a predetermined level. After that, mathematical expression of per-class centroids with present sensing result related to class l is given in equation (10):

$$\mu_j = \frac{1}{C_l} \sum_{p \in C_j} a_p \quad (10)$$

According to Equation (10), a_p denotes the collection of sample indices that correspond to class l . Local choice for p th CR user for l th class represented by S_{pl} is given in eq. (11) in accordance with the projected function.

$$S_{pV} = \operatorname{argmin}_{j \in B} (\mu_l - a) \quad (11)$$

Equation (11), which provides the expected function, predicts that the SUs will utilise the licenced spectrum whenever they are idle after monitoring PU activity and identifying potential attackers.

Reinforcement multilayer Q-graph colony optimization:

Assume a CRN with $|N|$ uniformly distributed SUs and $|M|$ uniformly distributed PUs, where $|N|$ is set of SU nodes and $|M|$ is set of PU nodes. Consider that there are numerous source SUs that are producing data packets to deliver in a multi-hop fashion through intermediary SUs to the destination SU node.

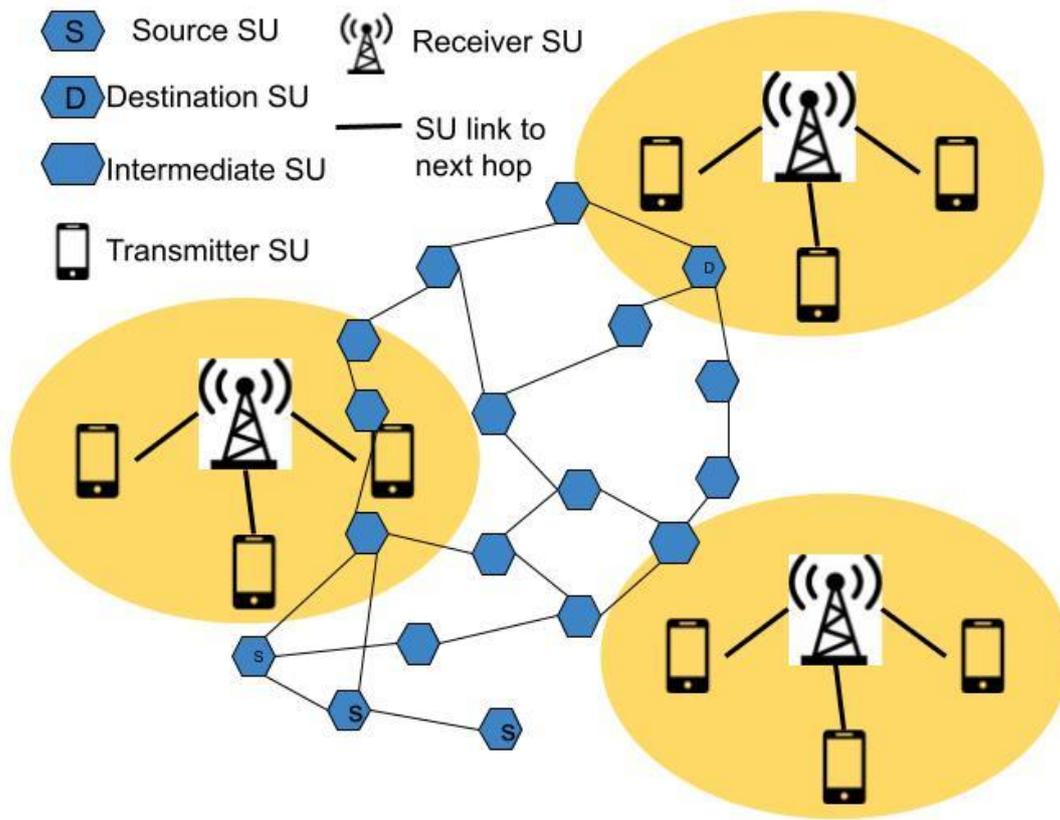


Fig 3 multihop routing architecture in proposed CRN networks

Every transmitter PU can relay data streams from other transmitter PUs, as shown in Fig. 3. Additionally, each transmitter PU can generate several data streams, which receiver PU can receive. Because it is adapted to data traffic in a way that closely resembles the LRD behaviour by several orders of magnitude, an MMPP is a useful method for simulating data traffic. DT-MMPP time steps in the PUs and many time steps in data transmission structure of SUs are synchronised. Additionally, we presumptively can simulate the state change of arriving traffic from other SUs for every SU using Markovian models. For the purpose of attaining expected restricted cost capability, the SU hubs require state progress likely capabilities. Following equation is then used to update the Q-value for the subsequent time step. (12)

$$Q_i^{t+1}(s_i, a_i) = (1 - v(t))Q_i^t(s_i, a_i) +$$

$$v(t) \left(\sum_{a_{-i}} (W_i(s_i, a_i, a_{-i}) + L_i(s_i, a_i, a_{-i})) \prod_{k \in I_j \setminus \{i\}} \pi_k^t(s_k, a_k) + \sum_{b_{-i}} Z_i(s_i, a_i, b_{-i}) \prod_{j \in H_i \setminus \{i\}} \pi_j^t(s_j, b_j) \right) + \beta \min_{c_i} Q_j^t(s'_i, c_i) \quad (12)$$

According to (12), since SU i's average one-hop delay is dependent on the actions of SUs in its neighbourhood, SU i needs their strategies in order to update Q-value. Additionally, for SU i to update Q-value, SUs in H_i 's strategies are required. But the problem is that SU i is unaware of tactics used by nodes in I_i and H_i , and getting this knowledge from other SUs through data exchange violates non-cooperative aspect of routing issue by

increasing network overhead. In following, we offer a conjecture-based solution to this issue that does not call for communication between contending SUs.

SU i does not know values $f_i^t(s_i, a_{-i}) = \prod_{k \in I_j \setminus \{i\}} \pi_k^t(s_k, a_k)$ and $g_i^t(s_i, b_{-i}) = \prod_{j \in H_i \setminus \{i\}} \pi_j^t(s_j, b_j)$ to update Q-value, but it can evaluate these values. To evaluate $f_i^t(s_i, a_{-i})$ and $g_i^t(s_i, b_{-i})$ SU i requires to maintain two functions $q_{1,i}(s_i, a_i)$ and $q_{2,i}(s_i, a_i)$ updated utilizing following eq. (13,14)

$$q_{1j}^{t+1}(s_i, a_i) = (1 - v(t))q_{1j}^t(s_i, a_i) + v(t) \left(\sum_{a_{-i}} (W_i(s_i, a_i, a_{-i}) + L_i(s_i, a_i, a_{-i})) f_i^t(s_i, a_{-i}) + \beta \min_{c_i} q_{1j}^t(s'_i, c_i) \right) \quad (13)$$

$$q_{2j}^{t+1}(s_i, a_i) = (1 - v(t))q_{2j}^t(s_i, a_i) + v(t) \left(\sum_{b_{-i}} Z_i(s_i, a_i, b_{-i}) g_i^t(s_i, b_{-i}) + \beta \min_{c_i} q_{2j}^t(s'_i, c_i) \right) \quad (14)$$

Where $f(s_i, a_{-i})$ and $g(s_i, b_{-i})$ are, respectively, estimates of $f(s_i, a_{-i})$ and $g(s_i, b_{-i})$. In accordance with equations (7) and (8), $q_{1,i}(s_i, a_i)$ is updated utilizing average one hop delay plus interference cost of SU i, and $q_{2,i}(s_i, a_i)$ is updated utilizing delay of a chosen next-hop to destination that are fed back to SU i. The following equation (15, 16) can be written using the Boltzmann distribution.

$$\psi_{1,i}^t(s_i, a_i) = \frac{e^{-q_i^t(s_i, a_i)/\tau}}{\sum_{c \in A_i} e^{-q_i^t(s_i, c)/\tau}} \quad (15)$$

$$\psi_{2,i}^t(s_i, a_i) = \frac{e^{-q_{2,i}^t(s_i, a_i)/\tau}}{\sum_{c \in A_i} e^{-q_{2,i}^t(s_i, c)/\tau}} \quad (16)$$

Where τ is a positive parameter. Then, evaluations of $f_i^t(s_i, a_{-i})$ and $g_i^t(s_i, b_{-i})$ are updated as eq. (17,18)

$$\tilde{f}_i^t(s_i, a_{-i}) = \tilde{f}_i^{t-1}(s_i, a_{-i}) - u_i^{s_i, a_{-i}} [\psi_{1,i}^t(s_i, a_i) - \psi_{1,i}^{t-1}(s_i, a_i)] \quad (17)$$

$$\tilde{g}_i^t(s_i, b_{-i}) = \tilde{g}_i^{t-1}(s_i, b_{-i}) - u_i^{s_i, b_{-i}} [\psi_{2,i}^t(s_i, a_i) - \psi_{2,i}^{t-1}(s_i, a_i)] \quad (18)$$

Where $u_i^{s_i, a_{-i}} > 0$ and $u_i^{s_i, b_{-i}} > 0$ are conjecture belief factors. The formula below updates Q-value for SU i based on estimates derived in (19).

$$Q_i^{t+1}(s_i, a_i) = (1 - v(t)Q_i^t(s_i, a_i) + v(t) \left(\sum_{a_{-i}} (W_i(s_i, a_i, a_{-i}) + L_i(s_i, a_i, a_{-i})) \tilde{f}_i^t(s_i, a_{-i}) + \sum_{b_{-i}} Z_i(s_i, a_i, b_{-i}) \tilde{g}_i^t(s_i, b_{-i}) \right) + \beta \min_{c_i} Q_i^t(s_i, c_i)) \quad (19)$$

The Boltzmann distribution is used to derive method of SU i in response to methods of other SUs from the following equation (20).

$$\pi_i^t(s_i, a_i) = \frac{e^{-Q_i^t(s_i, a_i)/\tau}}{\sum_{c \in A_i} e^{-Q_i^t(s_i, c)/\tau}} \quad (20)$$

When collecting characteristics during the graph convolution calculation, nodes with an excessive number of neighbours will have a significant gap with other nodes. We define standardised adjacency matrix as $A_{i,j} = D^{-1/2} A_{i,j} D^{-1/2}$ in order to get around this issue. Two graph convolutional layers later, eq. (21) yields output tensor H.

$$H = \text{ReLU} \left(\tilde{A}_{i,j} \left(\text{ReLU} \left(\tilde{A}_{i,j} X W_g^{(0)} + b_g^{(0)} \right) \right) W_g^{(1)} + b_g^{(1)} \right) \quad (21)$$

Where $W_g^{(0)} \in R^{F \times C}$, $b_g^{(0)} \in R^{C \times 1}$, $W_g^{(1)} \in R^{C \times Z}$ are weights that required to be trained in Actor method. The distinctive dimensions of output following graph convolution are C and Z. One-hot codes, which stand in for distinct ID of each node, are the building blocks of both vectors. You can use (22) to get state vector of current environment S.

$$S = \text{Concatenate}(\text{Flatten}(H), l, d) \quad (22)$$

Fully connected NN of Actor method will be fed state vector S as its input, and its output will be the matching action vector R N1. Action vector output by Actor fully connected layer must be filtered using mask vector because network architecture is typically not fully linked. Adjacency data of node where packet is situated is represented by the vector mask. Eq. (23) can be used to compute it.

$$\text{mask} = A_{i,j}^T l \quad (23)$$

where $A_{i,j}^T$ is adjacency matrix's transpose. Setting mask vector has two main goals: to reduce the decision space and prevent data packets from being forwarded to nodes that are not neighbours. The Actor model's final output can be generated using equation (24)

$$\mu = \text{softmax}(\text{mask} \odot (W_a S + b_a)) \quad (24)$$

where W_a, b_a are weights of completely connected layer and stands for the Hadamard product. The objective function is utilised to identify the particles by framing fitness value as a pop particle. Population is considered to be composed of PParticles, and the FParticle is utilised to evaluate the fitness value. The random creation of the target value, the FParticle, and termination in target condition in PSO all contribute to velocity.

1. The particle mentions dimension with position as well as average length travelled and is considered to be from the cluster head. 2. Based on the population's size, random distribution is used. 3. Using the Euclidean distance equation (25), path distance is calculated based on fitness value at each node.

$$\text{PD} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (25)$$

4. To achieve intended outcome in particle estimate with the least amount of aggregation, new particle is produced from previously determined fitness value. (i) According to the specified manipulation, as shown in eq. (26) velocity is achieved in particle location with regard to rate and particle position is modified correspondingly.

$$\text{Vel}_N = \rho * \text{Vel}_{\text{old}} + \rho_1(\text{FParticle} - \text{CP Particle}) + \rho_2(\text{FParticle} - \text{CP Particle}) \quad (26)$$

According to equation (27) in the PSO node, the new location is calculated using old value and velocity.

$$\text{Position}_N = \text{OLD}_N + \text{Vel}_N \quad (27)$$

So, the next particle travels with measured velocity and position when it arrives. 5. Fitness value is calculated using position's new value. 6. Comparison of the new and old fitness values is used to determine the next path, and the new position is used by equation (28), which specifies the measured iteration process.

$$\text{if } FN_{\text{NEW}} < FN_{\text{OLD}} FN_{\text{OLD}} = FN_{\text{NEW}} \quad (28)$$

7. Best inter-cluster aggregation path is supposed to be iteration in which highest fitness value obtained of best fitness.

Proposed routing algorithm:
Input: batch size, θ^{μ^*}
Load weights: θ^{μ^*} for target Actor method
Produce data packet p , utilizing θ^{μ^*} and stochastic OU process to find its next hop a
if p has arrived at its destination then done = True
else
observe network to get new state, evaluate reward r for $p \rightarrow a$ and append experience replay
$p \rightarrow \text{state} := \text{newstate}; p \rightarrow \text{location} := \text{location}$
list with $\{p \rightarrow \text{state}, p \rightarrow a, r, \text{newstate}, \text{done}\}$
if not done then
use θ^{μ^*} to find its next hop a
$p \rightarrow a := a$
send data packet p
end if

4. Performance analysis:

The proposed model is implemented in the network simulator NS2. The simulation is performed for the mobile nodes in the region coverage of $1000 \text{ m} \times 1000 \text{ m}$ with the standard distance of 1 Km for the total simulation time of 100 seconds. The node mobility for the movement is considered in the range of 10 m/sec to 50 m/sec. Every node in the network exhibits the equal transmission range of 250 meters. The node count is varies between 50 – 100 nodes. The node transmission power is stated as 0.660W and receiving power of nodes are defined as 0.395W with the initial energy level of 100J.

where A is size of bounded domain G and G is a plane-bound domain in our simulation process. Network topology is depicted in Figure 4. Figure 4a depicts route that source node, SU node n24, takes via DQN to the destination node, SU node n10. Figure 5b depicts SU node n22 as a newly joined source node that learns routing protocol from expert source node n24 and routes messages. Relay nodes n0 and n6 have both moved, and topology of the network has changed. Additionally, SUs have access to ten licensed channels and forty transmit power levels: 100, 120, 140, . . . , 1000 mW}

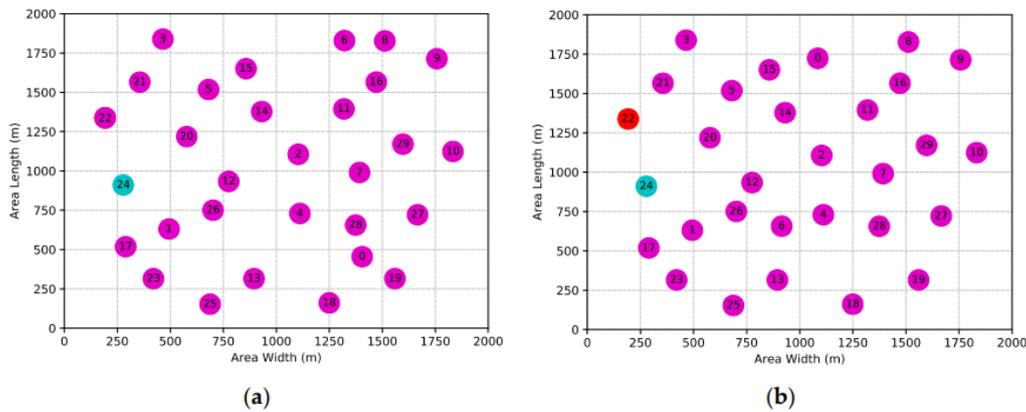


Fig 4. Network topology consisting of 30 SUs: (a) Route establishment by self-learning, and (b) route establishment by apprenticeship learning.

The assessment is done on a network that is $1000 \times 1000 \text{ m}$ wide and has packet sizes of 2500 bytes 500 broadcast

packets per session is the maximum allowed. Total spectrum will be accessible in the range of 54 to 72

MHz. The only available bandwidth for CRs is 2, 4, and 6 MHz. The CCC has a 2 MHz bandwidth. The simulation results show that the feasible target of our distributed method is near to upper bound. Since the ideal arrangement (obscure) lies between upper bound as well as possible arrangement got by our circulated calculation, we infer that outcomes acquired by our appropriated calculation are significantly nearer to the ideal arrangement and are subsequently profoundly serious.

We take into account 100 distinct network instances in this study. In a 100x100 area, every network has either $|N| = 20, 30, 40,$ or 50 nodes distributed at random. The appropriate dimensions are used to normalize the distance, rate, and power density units. The network is composed of $|M| =$ ten distinct frequency bands. However, only a subset of these frequency bands might be accessible at each node. There are $|L| = 3$ or 5 sessions among these nodes; the sessions' source and destination nodes are chosen at random, minimum rate needs is chosen at random within $[1, 10]$. Each node's maximum transmission range is assumed to be 20.

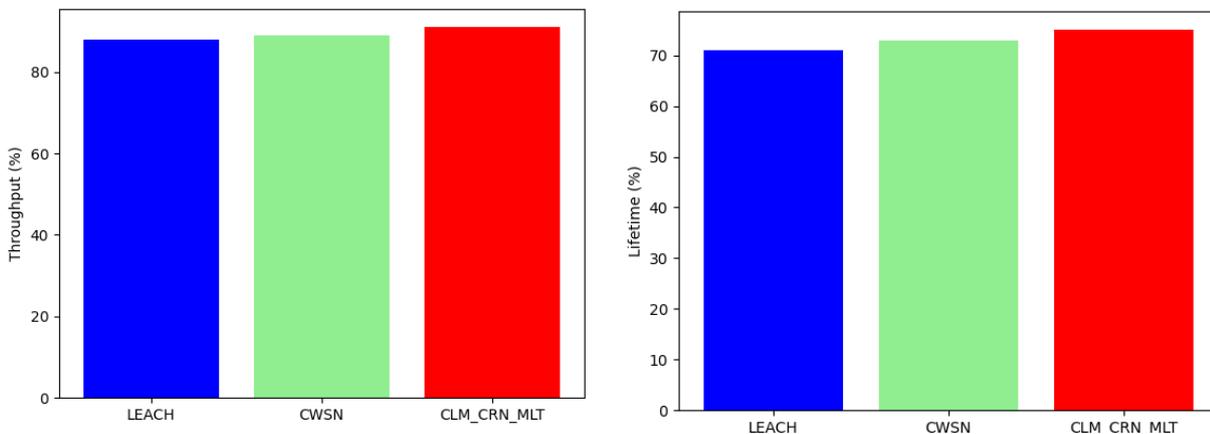
CRN analysis based on proposed model for carious network scenario:

Table-1 Comparative analysis based on various network topology between proposed and existing technique

Dataset	Techniques	Throughput	Lifetime	Jamming prediction	Energy efficiency	Routing latency	PDR
PHY layer	LEACH	88	71	59	77	39	91
	CWSN	89	73	61	79	42	93
	CLM_CRN_MLT	91	75	63	81	43	95
Peer-to-peer topology nodes	LEACH	91	72	62	79	45	92
	CWSN	93	75	65	83	49	94
	CLM_CRN_MLT	95	76	66	85	51	96

Table-1 shows analysis based on various network topology. Here network topology compared are PHY layer and Peer-to-peer topology nodes in terms of throughput, lifetime, jamming prediction, energy efficiency, routing latency, PDR.

We discovered that the model offered an optimized configuration to us. The model calculates these PHY layer output configuration parameters in accordance with the application's requirements. Figure 5 depicts them on the source node. In this case, the primary constraint imposed by the application is that the delay must not be longer than one millisecond and that the BER must be less than or equal to 103. The model estimates that the transmission power is 5 watts, so the appropriate modulation scheme, 16DPSK in this instance, is chosen. The problem is solved by the model by taking into account two main parameters: BER is supposed to be less than 103. SDR power will remain available for more pressing scenarios as a result of this. Method determines the PHY layer's output configuration parameter in accordance with application requirements. Table 4 shows them as a node. The application's fundamental restriction in this case is that the data rate must strictly be less than or equal to 8 kbps, and delay can be tolerated up to 2 ms/2 hops. In this instance, where mobile SDRs have moved far apart, the PHY layer can be set up in two different ways depending on the application's needs.



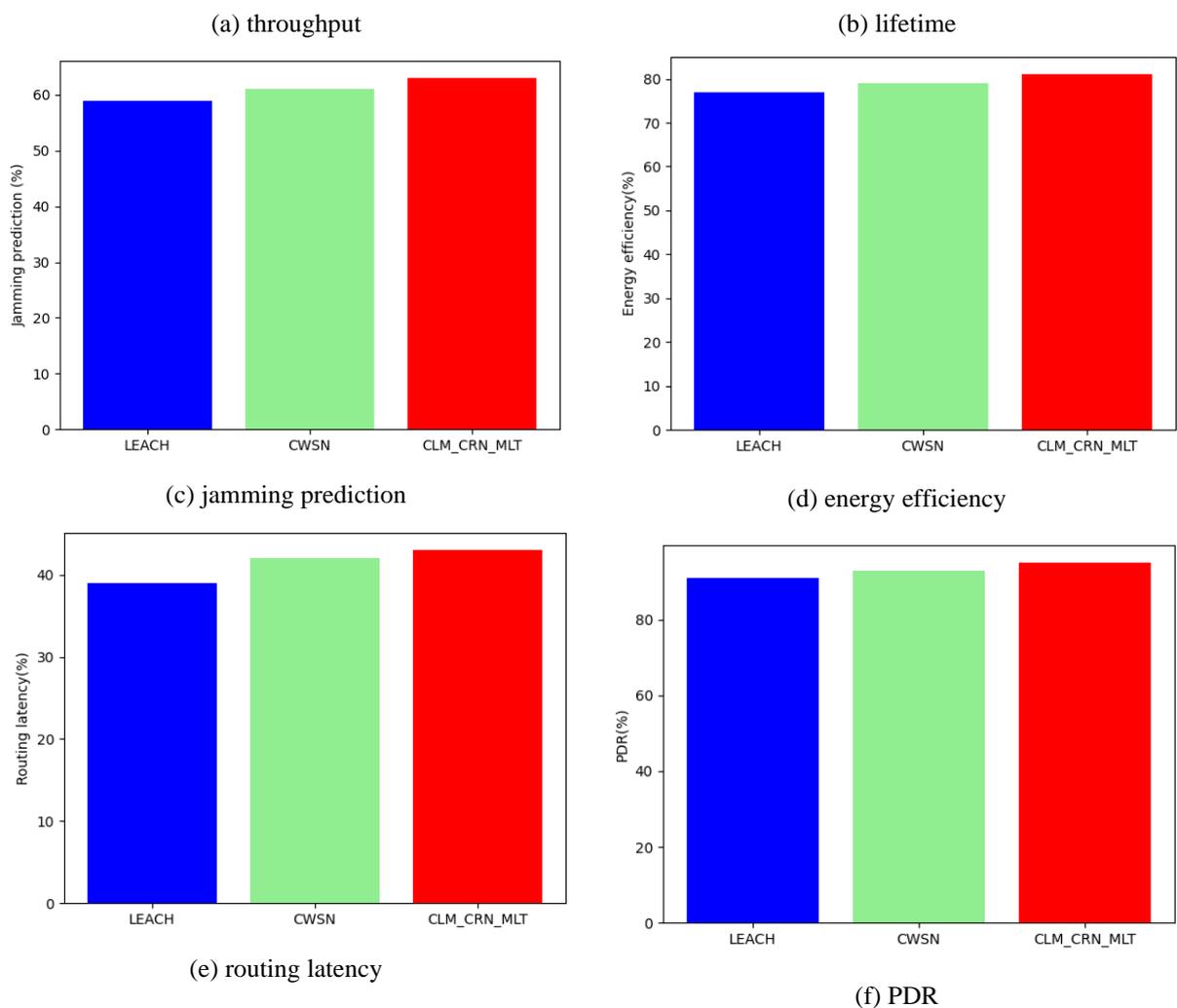
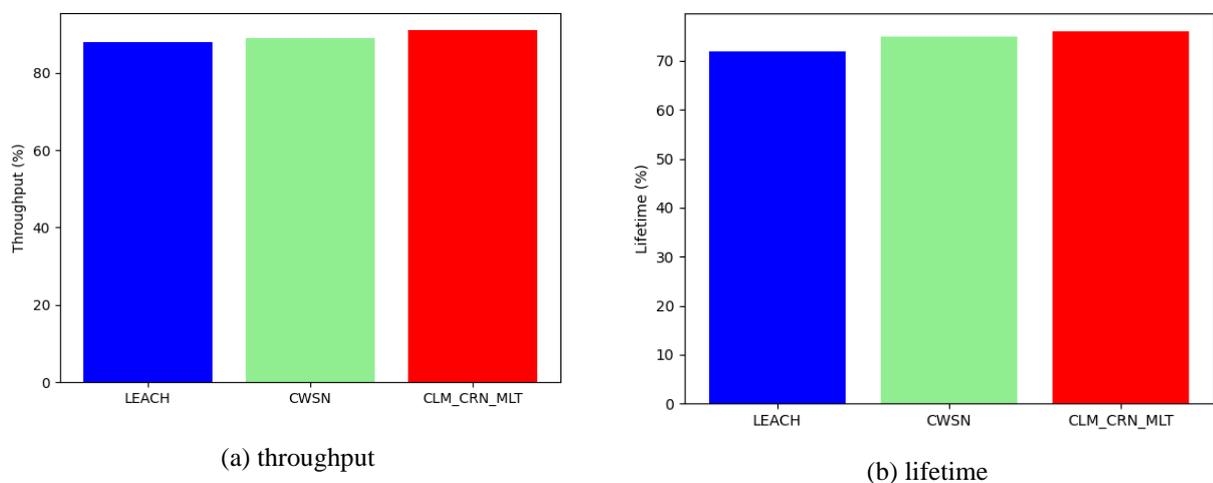


Fig-5 analysis based on PHY layer network model in terms of (a) throughput,(b) lifetime,(c) jamming prediction,(d) energy efficiency,(e) routing latency,(f) PDR

The above figure 5 (a)- (f) shows comparison based on PHY layer network analysis. The proposed technique throughput of 91%, lifetime of 75%, jamming prediction of 63%, energy efficiency of 81%, routing latency of 43%, PDR of 95%; existing LEACH throughput of 88%,

lifetime of 71%, jamming prediction of 59%, energy efficiency of 77%, routing latency of 39%, PDR of 91%, CWSN attained throughput of 89%, lifetime of 73%, jamming prediction of 61%, energy efficiency of 79%, routing latency of 42%, packet delivery ratio of 93%.



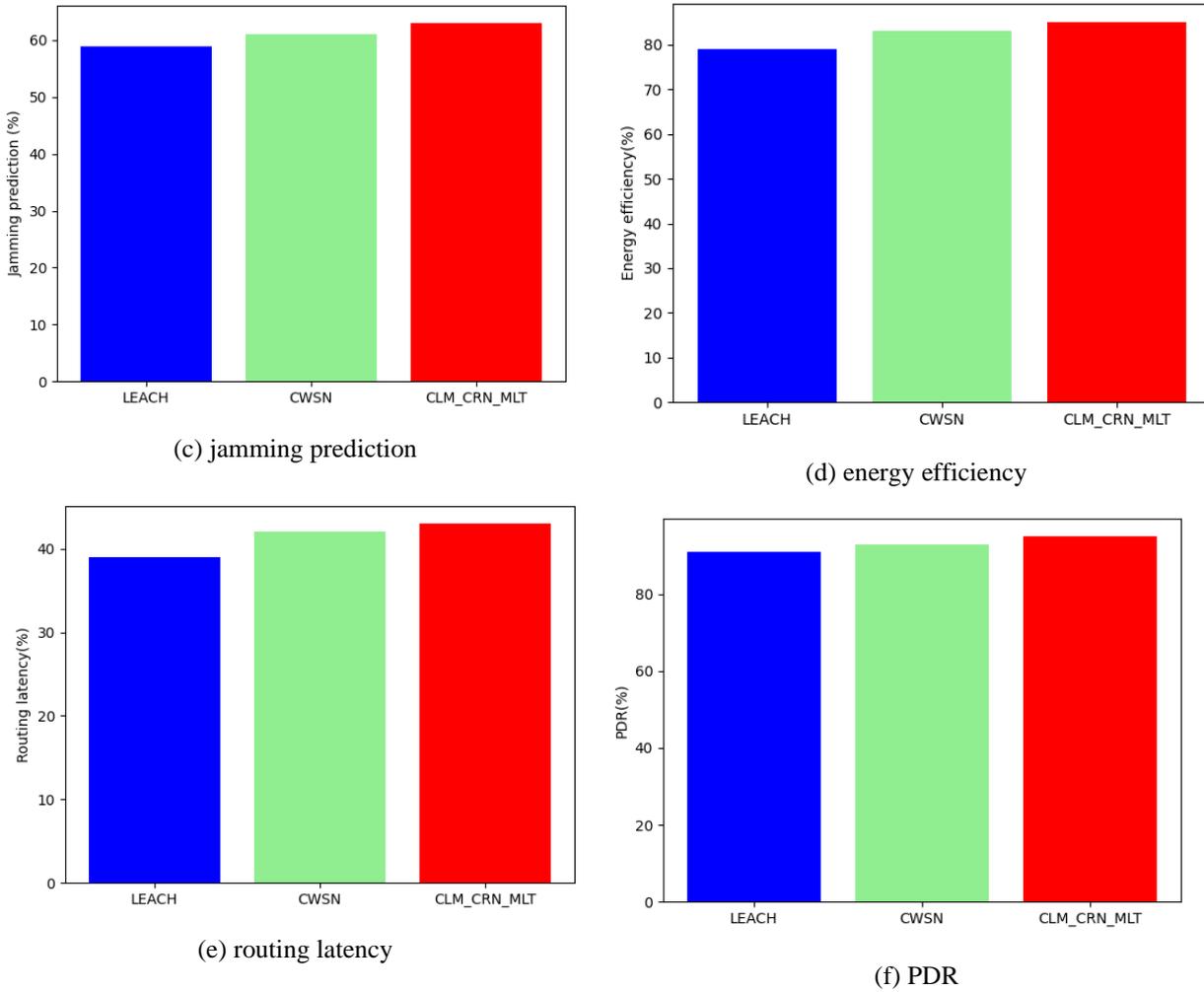


Fig-6 analysis based on Peer-to-peer topology nodes network model in terms of (a) throughput,(b) lifetime,(c) jamming prediction,(d) energy efficiency,(e) routing latency,(f) PDR

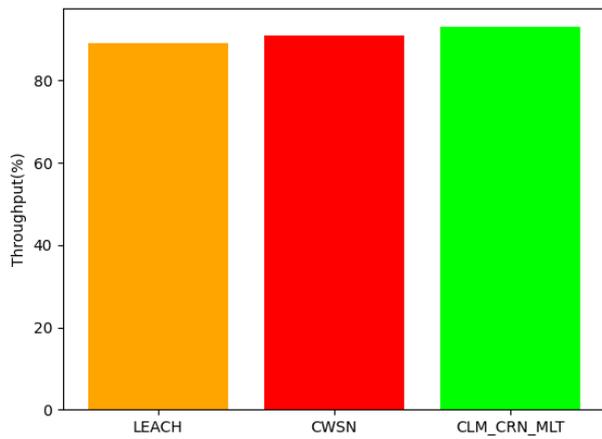
Figure 6 (a)- (f) analysis for Peer-to-peer topology nodes network model is shown. Here proposed technique attained throughput of 95%, lifetime of 76%, jamming prediction of 45%, PDR of 92%, CWSN throughput of 93%, lifetime of 75%, jamming prediction of 62%, energy efficiency of 85%, routing latency of 51%, PDR of 96%; existing LEACH attained

throughput of 91%, lifetime of 72%, jamming prediction of 62%, energy efficiency of 79%, routing latency of 45%, PDR of 92%.

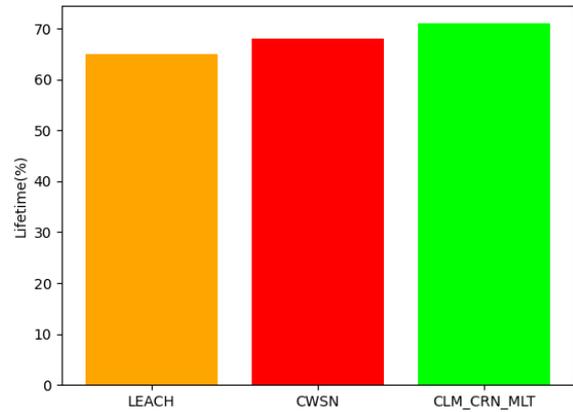
Table-2 analysis based on number of CRN nodes and number of episodes

Network parameters	Techniques	Throughput	Lifetime	Jamming prediction	Energy efficiency	Routing latency	PDR
number of CRN nodes	LEACH	89	65	75	55	45	81
	CWSN	91	68	76	59	49	83
	CLM_CRN_MLT	93	71	79	61	51	85
Number of episodes	LEACH	92	68	77	59	52	85
	CWSN	95	72	79	63	53	86
	CLM_CRN_MLT	96	73	82	65	55	88

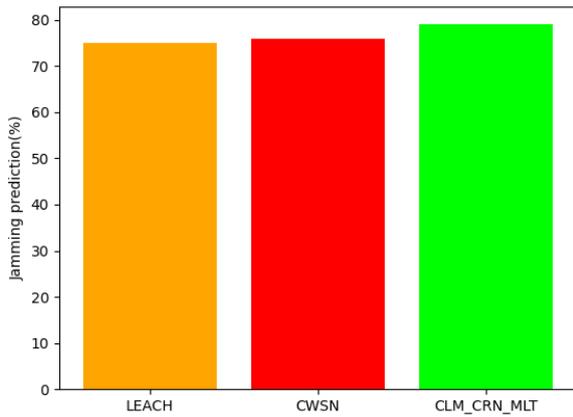
Table-2 shows analysis based on number of CRN nodes and number episodes. Here parameters analysed are throughput, lifetime, jamming prediction, energy efficiency, routing latency, PDR.



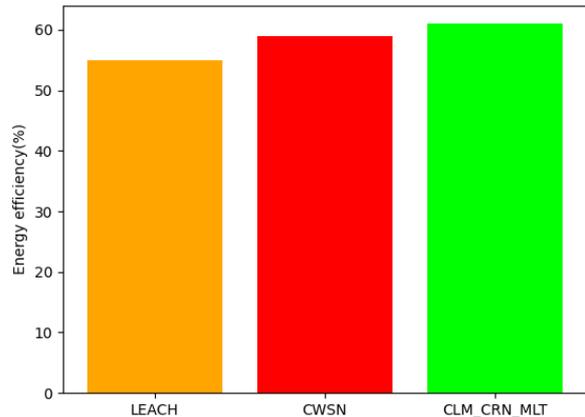
(a) throughput



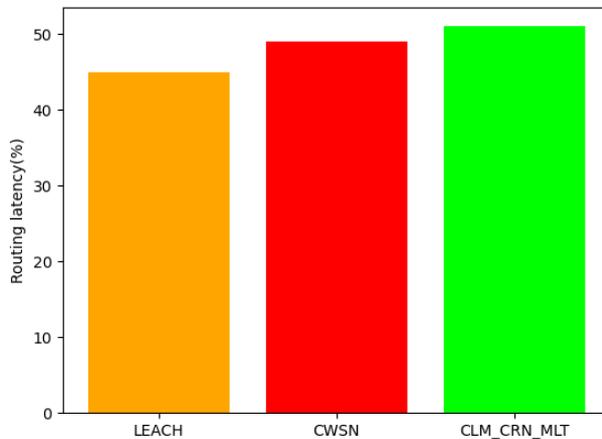
(b) lifetime



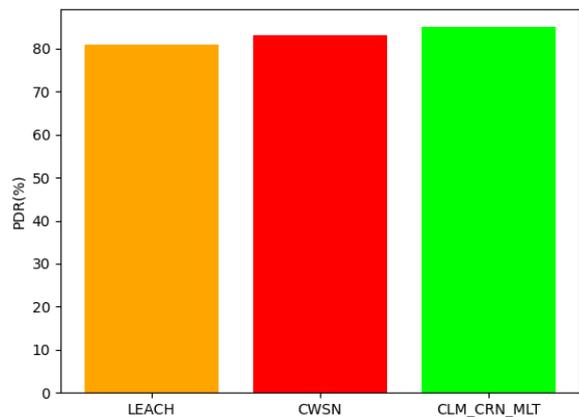
(c) jamming prediction



(d) energy efficiency



(e) routing latency



(f) PDR

Fig-7 analysis based on number of CRN nodes in terms of (a) throughput, (b) lifetime, (c) jamming prediction, (d) energy efficiency, (e) routing latency, (f) PDR

Figure 7 (a) - (f) shows comparison based on number of CRN nodes. Proposed technique attained throughput of 93%, lifetime of 71%, jamming prediction of 79%, energy efficiency of 61%, routing latency of 51%, PDR of 85%; existing LEACH throughput of 89%, lifetime of

65%, jamming prediction of 75%, energy efficiency of 55%, routing latency of 45%, PDR of 81%, CWSN attained throughput of 91%, lifetime of 68%, jamming prediction of 76%, energy efficiency of 61%, routing latency of 51%, PDR of 85%.

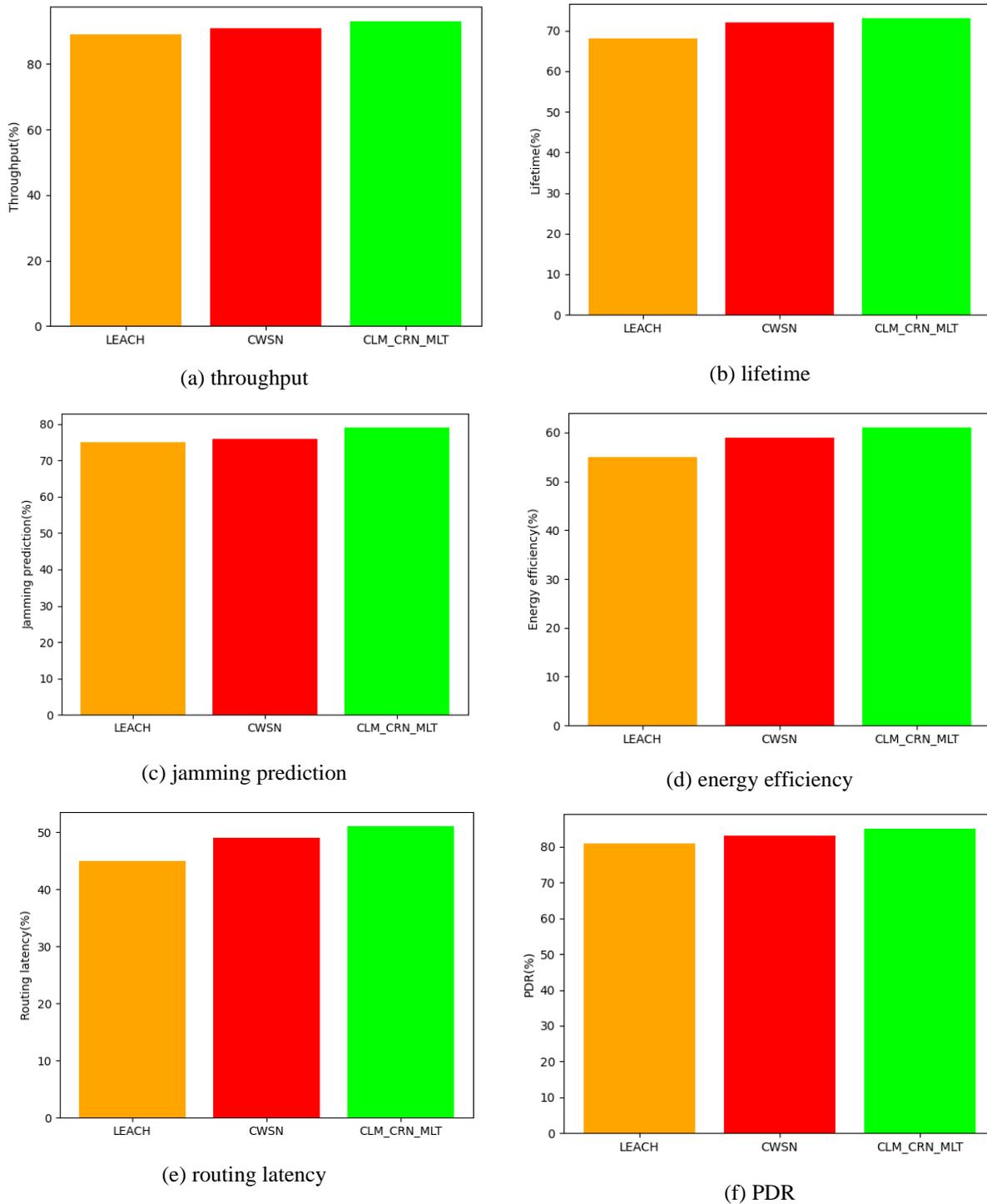


Fig-8 analysis based on number of episodes in terms of (a) throughput,(b) lifetime,(c) jamming prediction,(d) energy efficiency,(e) routing latency,(f) PDR

Figure 8(a)- (f) analysis for Number of episodes network model is shown. Here proposed technique attained throughput of 96%, lifetime of 73%, jamming prediction of 82%, energy efficiency of 65%, routing latency of 55%, PDR of 88%; existing LEACH attained throughput of 92%, lifetime of 68%, jamming prediction of 77%, energy efficiency of 59%, routing latency of 52%, PDR of 85%, CWSN attained throughput of 95%, lifetime of 72%, jamming prediction of 79%, energy efficiency of

63%, routing latency of 53%, packet delivery ratio of 86%.

The network's parameter settings are shown in Table 3. In order to maximize the agent's performance in our problem environment, we evaluate a few hyperparameters of the reinforcement implementation. In particular, we take into account the three hyperparameters listed below: the ratio of actor model's a_lr learning rate, critic model's c_lr learning rate, and

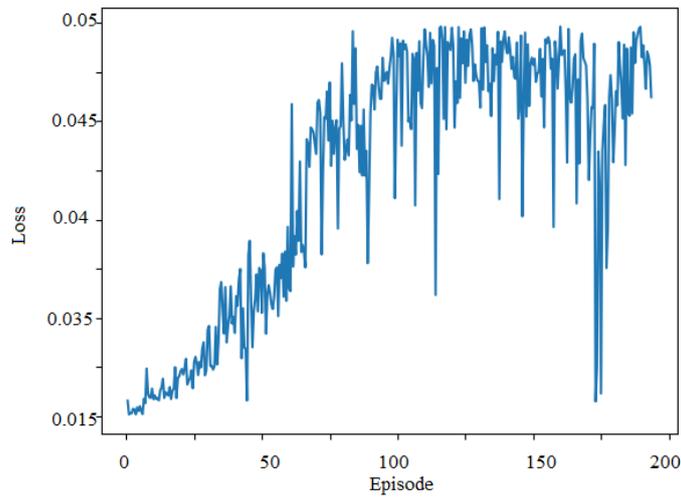
the soft replacement TAU. Actor as well as Critic model are described in detail in Table 4.

Table 3. Hyperparameters of reinforcement learning.

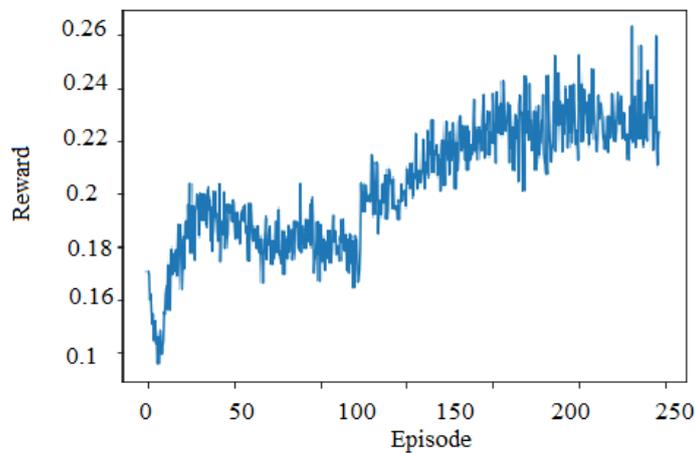
Parameters	Setting
Learning rate for Actor method	$a_lr = 0.0001$
Reward discount	$\gamma = 0.95$
Soft replacement	$\tau = 0.1$
Learning rate for Critic method	$c_lr = 0.001$
Final random exploration rate	Final_epsilon= 0.01
Initial random exploration rate	Initial_epsilon= 1.0

Table 4. Details of method struct.

Model Name	Layer Name	Parameter Details		
		Hidden Units	Activation	Trainable Weights
Actor model	256	ReLU		
	GCN1	8	ReLU	
	GCN2	64	ReLU	10,774
	Denses1	N	Linear	
	Denses2	1	Softmax	
	Output	256	ReLU	
	GCN1	8	ReLU	10,113
	GCN2	64	ReLU	
	Dense layer	1	Linear	



(a) Loss



(b) Reward

Fig 9. Total loss and reward trend on model training process (a) loss. (b) reward

Method is utilized to predict transmission direction in simulation environment following pre-training. Experience pool will house the environment data as well as the associated reward value. Experience playback as well as gradient descent will be used to update the model's neural networks' weights. Each 100-step iteration of loop will be repeated 200 times. Throughout the training process, the loss and reward are depicted in Figure 9. The misfortune bend shows a descending pattern overall. However, during the training, the loss frequently rises. When a new state space with better reward is encountered, loss curve changes. Because controller doesn't know enough about network as well as explores environment, reward is lower at first. The reward quickly rises after a few training sessions. It demonstrates that the controller's routing policies can direct packet forwarding for improved returns.

5. Conclusion:

This research propose novel technique in transmission control with routing optimization using machine learning

in cognitive radio networks using cross layer model. The spectrum sensing (SS) process that finds whether a PU signal is present or absent consumes a significant amount of energy and time in wireless nodes, reducing both SU throughput and battery power. We propose a centralised training and distributed execution architecture that separates method training as well as inference processes and enables intelligent routing decision-making for resource-constrained nodes with little computational complexity. In order to extract fine-grained data from our hybrid routing metric, we additionally develop a special feature creation and extraction method as well as a new hybrid routing measure. The recommended CDRL strategy, which uses a collaborative and iterative model optimisation technique, yields the greatest long-term network performance. Proposed technique attained throughput of 96%, lifetime of 73%, jamming prediction of 82%, energy efficiency of 65%, routing latency of 55%, packet delivery ratio of 88%; existing LEACH attained throughput of 92%, lifetime of 68%, jamming prediction of 77%, energy efficiency of 59%, routing

latency of 52%, PDR of 85%, CWSN attained throughput of 95%, lifetime of 72%, jamming prediction of 79%, energy efficiency of 63%, routing latency of 53%, packet delivery ratio of 86%.

Reference:

- [1] Wang, Y., Shang, F., & Lei, J. (2023). Energy-efficient and delay-guaranteed routing algorithm for software-defined wireless sensor networks: A cooperative deep reinforcement learning approach. *Journal of Network and Computer Applications*, 103674.
- [2] Huang, R., Guan, W., Zhai, G., He, J., & Chu, X. (2022). Deep graph reinforcement learning based intelligent traffic routing control for Software-Defined wireless sensor networks. *Applied Sciences*, 12(4), 1951.
- [3] Duong, T. V. T., & Binh, L. H. (2022). IRSML: An intelligent routing algorithm based on machine learning in software defined wireless networking. *ETRI Journal*, 44(5), 733-745.
- [4] Malik, T. S., Malik, K. R., Afzal, A., Ibrar, M., Wang, L., Song, H., & Shah, N. (2022). RL-IoT: Reinforcement learning-based routing approach for cognitive radio-enabled IoT communications. *IEEE Internet of Things Journal*, 10(2), 1836-1847.
- [5] Dhiman, G., & Sharma, R. (2022). SHANN: an IoT and machine-learning-assisted edge cross-layered routing protocol using spotted hyena optimizer. *Complex & Intelligent Systems*, 8(5), 3779-3787.
- [6] Alqahtani, A. S., Changalasetty, S. B., Parthasarathy, P., Thota, L. S., & Mubarakali, A. (2023). Effective spectrum sensing using cognitive radios in 5G and wireless body area networks. *Computers and Electrical Engineering*, 105, 108493.
- [7] Farquhar, C., Kafle, S., Hamedani, K., Jagannath, A., & Jagannath, J. (2023). Marconi-Rosenblatt Framework for Intelligent Networks (MR-iNet Gym): For Rapid Design and Implementation of Distributed Multi-agent Reinforcement Learning Solutions for Wireless Networks. *Computer Networks*, 222, 109489.
- [8] Gupta, A., & Joshi, B. K. (2023). Efficient Optimized ATSDERP Routing Based DEQL Spectrum Sharing HPNCS Network Coding Model in Cognitive Radio Networks. *Wireless Personal Communications*, 129(4), 2995-3022.
- [9] Paul, A., & Maity, S. P. (2022). Reinforcement Learning Based Q-Routing: Performance Evaluation on Cognitive Radio Network Topologies. *Wireless Personal Communications*, 125(2), 1425-1441.
- [10] Tran, T. N., Nguyen, T. V., Shim, K., da Costa, D. B., & An, B. (2022). A Deep Reinforcement Learning-Based QoS Routing Protocol Exploiting Cross-Layer Design in Cognitive Radio Mobile Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 71(12), 13165-13181.
- [11] Sreeraj, A., Vijayalakshmi, P., & Rajendran, V. (2022, April). A Deep Learning Enabled Software-Defined Radio based Routing Protocol for Underwater Acoustic Sensor Networks. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)* (pp. 28-32). IEEE.
- [12] Li, J., Ye, M., Huang, L., Deng, X., Qiu, H., & Wang, Y. (2023). An Intelligent SDWN Routing Algorithm Based on Network Situational Awareness and Deep Reinforcement Learning. *arXiv preprint arXiv:2305.10441*.
- [13] Jagannath, J., Jagannath, A., Henney, J., Gwin, T., Kane, Z., Biswas, N., & Drozd, A. (2022). Design of fieldable cross-layer optimized network using embedded software defined radios: Survey and novel architecture with field trials. *Computer Networks*, 209, 108917.
- [14] Kumar, S., & Raja, P. (2022). Implementation of Artificial Intelligence in a Software-Defined Wireless Sensor Network. *Journal of Artificial Intelligence, Machine Learning and Neural Network (JAIMLNN) ISSN: 2799-1172*, 2(06), 32-42.
- [15] Natarajan, Y., Srihari, K., Dhiman, G., Chandragandhi, S., Gheisari, M., Liu, Y., ... & Alharbi, H. F. (2022). An IoT and machine learning-based routing protocol for reconfigurable engineering application. *IET Communications*, 16(5), 464-475.
- [16] Quan, W., Liu, M., Cheng, N., Zhang, X., Gao, D., & Zhang, H. (2022). Cybertwin-driven DRL-based adaptive transmission scheduling for software defined vehicular networks. *IEEE Transactions on Vehicular Technology*, 71(5), 4607-4619.
- [17] Paul, A., & Choi, K. (2023). Joint spectrum sensing and D2D communications in Cognitive Radio Networks using clustering and deep learning strategies under SSDF attacks. *Ad Hoc Networks*, 143, 103116.

- [18] Kafetzis, D., Vassilaras, S., Vardoulas, G., & Koutsopoulos, I. (2022). Software-defined networking meets software-defined radio in mobile ad hoc networks: State of the art and future directions. *IEEE Access*, 10, 9989-10014.
- [19] Giri, M. K., & Majumder, S. (2022). Deep Q-learning based optimal resource allocation method for energy harvested cognitive radio networks. *Physical Communication*, 53, 101766.
- [20] Sixu, L., Muqing, W., & Min, Z. (2022). Particle swarm optimization and artificial bee colony algorithm for clustering and mobile based software-defined wireless sensor networks. *Wireless Networks*, 28(4), 1671-1688.
- [21] Rapetswa, K., & Cheng, L. (2023). Towards a multi-agent reinforcement learning approach for joint sensing and sharing in cognitive radio networks. *Intelligent and Converged Networks*, 4(1), 50-75.
- [22] G. M. ., Deshmukh, P. ., N. L., U. K. ., Macedo, V. D. J. ., K B, V. ., N, A. P. ., & Tiwari, A. K. . (2023). Resource Allocation Energy Efficient Algorithm for H-CRAN in 5G. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(3s), 118–126. <https://doi.org/10.17762/ijritcc.v11i3s.6172>
- [23] Kamau, J., Ben-David, Y., Santos, M., Joo-young, L., & Tanaka, A. Predictive Analytics for Customer Churn in the Telecom Industry. *Kuwait Journal of Machine Learning*, 1(3). Retrieved from <http://kuwaitjournals.com/index.php/kjml/article/view/130>
- [24] Verma, M. K., & Dhabliya, M. D. (2015). Design of Hand Motion Assist Robot for Rehabilitation Physiotherapy. *International Journal of New Practices in Management and Engineering*, 4(04), 07–11.
- [25] Dhabliya, M. D. (2019). Uses and Purposes of Various Portland Cement Chemical in Construction Industry. *Forest Chemicals Review*, 06–10.