International Journal of Global Science Research ISSN: 2348-8344 (Online)
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305
Available Online at www.ijgsr.com

Research Paper

Implementation of Gaussian Models for Improving Noise Pollution Efficiency In Cognitive Radio Networks

Syed Wahid Hasan[1*], Fauzia Sadaf [2], Arshi Rana[3], Amna Bakhtawar[4], Arshad Jamal[5], Suraiya Fatima[6], Nasim Equbal[7] and Susheel Kumar[8]
[1*]Post Graduate Department of Zoology, Magadh University, Bodh Gaya, Bihar, India
[2]RPS Mahila College, Patliputra University, Patna, Bihar, India
[3]Department of Zoology, Magadh Mahila College, Patna University, Patna, Bihar, India
[4]Department of Social Work, Jamia Millia Islamia University, New Delhi, India
[5]Department of Zoology, Soghra College, Biharsharif, Nalanda, Bihar, India
[6]Post Graduate Department of Environmental Science, Magadh University, Bodh Gaya, Bihar, India
[7]Post Graduate Department of Environmental Science, Magadh University, Bodh Gaya, Bihar, India
[8]Department of Zoology, Deen Bandhu Mahavidyalaya, Tenua, Basti, Uttar Pradesh, India
*Corresponding Author Email: wahidhasan00786@gmail.com

Received: 20/06/2025
Revised: 29/06/2025
Accepted: 05/07/2025

Abstract: Mobile communication has reduced the amount of spectrum available, but many of these resources are still underutilized. E2SS methodologies for cognitive radio (CR) systems based on machine learning are presented in this paper. Before it can classify, we have to teach the classifier. Unsupervised classification algorithms such as K-means segmentation and the Variational model are used in cooperative spectrum sensing. Mixture designs like Gaussian mixtures, on the other hand, are unable to understand the relationships between data when dealing with large dimensional data. This paper's responsive Gaussian Mixture Model utilizes a mixture of Gaussian distributions to model each data value. A Gaussian is a user feature. As the means and standard deviations of each distribution evolve, the algorithm gains a better understanding of the relationships between classes of channels, both available and unavailable. The weight and variance of each distribution are used to determine this. Using the expectation maximising algorithm for reducing dimension data features, this method renews the data group in the feature space. Each classification technique's precision, recollection, F1-measure, categorization error, ROC curve, likelihood of detection, and false detection probability are evaluated.

Keywords: Noice pollution, Cooperative spectrum sensing, K-means clustering.

Abbreviations: Cognitive Radio (CR), Cooperative Spectrum Sensing (CSS), Sequential Probability Ratio Test (SPRT)

Introduction:

The CR network is a new emerging technology that can address wireless communication's current issue is the inadequacy in the spectrum. Due to this, spectrum sensing will not directly contribute to increasing the CR's throughput. Alternatives to spectrum sensing may include spectrum prediction (Mitola, 2000 and Kolodzy 2002). It indicates wireless communications infrastructure that is capable of adaptively and autonomously changing its channel control conditions response to the electromagnetic environment in which it is used in operation. It is possible to utilize CSS when the CR devices are dispersed across various places. The CR devices may work together (Akyildiz et. al., 2011) to find a better solution to the hidden PU issue to enhance sensing reliability over individual sensing. CSS enhances a CR system's performance gain by allowing several CR users to work together to identify spectral gaps. While matched filtering outperforms other spectrum sensing methods like cyclostationary detection and energy detection, its complexity prevents it from being utilized in most systems. It is possible to discover primary user communications by utilizing the cyclostationarity features of the signals received (Khatereh and Jamshid, 2018; Lakshmanan and Hyils 2020) a technique for recognizing primary user communications. It uses the intervals of the signal source to decide whether or not primary users are present in the network. When used to detect primary, secondary, and interference users, this detector is capable of differentiating between them. In contrast, the achievement of this detection technique is dependent on an adequate sample size, which raises the computation cost of the method. It is possible to detect data singularities and edges using a wavelet transform. The wavelet-based spectrum access approach (Yongcheng et. al., 2019; Jongbu and Chang, 2020; Jayanthi et. al., 2020) commonly uses a train of consecutive regularity sub - band to disintegrate the bandwidths of interest (Yongcheng et. al., 2019; Jongbu and Chang, 2020; Jayanthi et. al., 2020). It is possible to detect oddities in these artists and determine whether continuum is full or empty using the wavelet transform. To improve spectrum sensing performance within a cognitive radio network, hybrid models were developed over the last few years. These models combine the strengths of two or more detection techniques.

The Bayesian approach and sequential probability ratio test (SPRT) are two of the most commonly used methods for energy detection in cognitive radio networks. They necessitate prior knowledge that isn't always available in a critical resource environment (CR), but they provide better protection for PUs and spectral hole exploration than current methods. Learning from their surroundings is the premise of cognitive radios. By monitoring their surroundings and making adjustments to their operational parameters (such as frequency and transmit power), users of CR are expected to do their part. To help CR users learn from their surroundings, researchers have studied machine learning algorithms (Han et. al., 2011; Davanam et. al., 2020; Bkassiny et. al., 2011; Sangamithra et. al., 2017; Reddy, 2008; Zhu, 2010). Spectrum sensing requires machine learning algorithms to extract the feature vector of an input a pattern and then classify it into a premise lesson that also indicates the exclusion or existence of PU exercise. They are able to resolve regression and classification issues because they are a member of the supervised learning method family. Training examples can be used to create K neighborhood classes in KNN (spectrum sensing feature vectors). CR users can use this algorithm because it has a low complexity for spectrum sensing.

In this paper, a machine learning-based CSS scheme is presented. The expectation maximization algorithm of a lowering dimension data feature is utilized when trying to train an Adaptive Gaussian mixture. Ensures the data cluster is updated in order the information is collected according to the attribute freedom. Feature vectors extracted from patterns and fed to a classification model, which assigns the trend to one of several classes, are commonly used in pattern classification by machine learning techniques. An advantage of this approach is that it can learn about its surroundings in an automated manner, including PU topology, cognitive radio networks and faded channels among other things.

Related works:

Due to the lack of information regarding PU signals, high computing costs, and performance limitations in low signal-to-noise ratio (SNR) conditions, the major constraints for most of these sensors are the inability to establish a threshold detection. A new detection technique based on machine learning was suggested by the authors in (Hassaan, 2019) to get over these restrictions. First, cyclostationary characteristics are used to identify anomalies. The low signal-to-noise ratio issue, determining the detection threshold, and high computing cost may all be addressed using an ensemble classifier. To begin, a dataset comprising several orthogonal frequency-division multiplexing signals with various signal-to-noise ratios (SNRs) is generated. To find cyclostationary features, the FFT accumulation method is employed. Classifier ensembles have been developed to notice PU's gesture in low-SNR situationS utilizing the retrieved features, as previously stated. The AdaBoost algorithm and decision trees form the basis of this ensemble classifier. Evidence suggests that the proposed classifier is superior to other machine learning classifiers, such as the svm algorithm (SVM). Simulation findings also indicate the detector proposed in this paper is more resilient and efficient than other current detectors like a cyclostationary detector without machine learning or a traditional energy detector.

Collaborative intellectual radio systems introduced a new trend recognition method for spectrum sensing in (Yasmin et. al., 2012). With this design, nodes collect discriminative features from received signals and utilize them in classifiers at central nodes to decide whether or not spectrum holes are available for exploitation by cognitive radio networks globally. Classifiers using energy, cyclostationary, or coherent features are suggested for both linear and polynomial classifiers. All proposed systems' simulation results, including detection and false alarm probabilities, are given. Cyclostationary-based methods are shown to be the most reliable for identifying primary spectrum users. This, however, necessitates a longer sensing time than what was previously possible with coherent-based strategies. Involvement in supplying features for the classifier improves performance, according to the findings. Due to the linear classifier's simplicity, it outperforms a second-order quadratic classifier in this application for spectrum sensing. Finally, we look at how the size of the observation period affects the detection rate.

However, there is still room for improvement in local sensing decisions to attain highly reliable spectrum sensing. According to (Waleed et. al., 2011) a two-stage local spectrum sensing method was suggested by the authors. As a first step, each cognitive radio will gather data using currently available spectrum access methods, such as detection method, corresponding filter sensing, and time frequency sensing. Fuzzy logic is used in the second step to combine the results from each of the techniques used in step 1 to conclude whether or not the primary user is there. The suggested method improves significantly in terms of sensing accuracy since low false alarms and a higher detection rate.

In order to reduce the cost of sensing and increase the amount of time available for data transmission, a proper prediction can identify PU activity quickly. The authors of Suddhendu et. al., 2019) developed a binary form of neural network (NN) prediction system to anticipate PU transmission activity. To check the routine of the projected approach, two important parameters are considered, and two methods are tested and compared: one is just prediction but no sensing; the other is the first prediction and then sensing. Variations of these parameters are assessed in traffic conditions passing through PU at any given time. Predictive sensing's results are compared to actual results.

In (Karaputugala et. al., 2013) the authors presented new spectrum sensing machine learning models for cognitive radio users. There are two approaches to classifying the data as the primary is K-means & Gaussian mixture model (GMM) that belongs to the unsupervised family. The secondary is SVM and KNN belongs to the supervised family. To determine if a radio channel vacancy and a classifier use the predicted energy levels from all the cognitive radios, treated as the feature vector. There are two types of feature vectors classified by the classifier: those that are "channel unavailable and unavailable classes. The proposed classifier is trained first before It is used to categorise content online.. K-means clustering is a classification method that identifies groups based on how many primary users (PUs) are in each one. The classifier uses this information to determine the class to which a particular test energy vector holds. There are Gaussian density functions generated by the GMM, which correctly represent the training data. Support vectors for SVM may be caused by increasing the margin between the model's separated

International Journal of Global Science Research ISSN: 2348-8344 (Online)
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305
Available Online at www.ijgsr.com

hyperplane and feature vectors. Classification performance is measured using metrics such as ROC contours, average session time, and latency.
In (Charan and Pandey, 2018), the instigators presented a channel sensing technique based on covariance, where the adaptive threshold is intelligently chosen to decrease the possibility of error even as still protecting the primary user (PU). It is essential to keep the overall decision error probability as minimal as possible by calculating an adaptive threshold that considers both detection probability and false alarm probability This adaptive threshold and two other threshold values based on CFAR and CDR techniques are fully considered in order to achieve the maximum protection against PU. In addition, the suggested method uses the minimum samples possible to provide reliable spectrum sensing. By means of exposure possibility and decision error likelihood, the proposed scheme outperforms current ED-based systems and the covariance-based detection technique, as shown by the simulation results.

Material and Method:

Multi SU cooperative sensing with several spectrum users is used to enhance the performance of this system. After collecting information regarding allowed channels, SUs send data to fusion centre (FC) via a reporting channel. The FC then performs a combined processing step and concludes. We illustrate the model of a cooperative spectrum sensing system in Figure 1 (below). If there are M SUs in a CRN collecting data, the data may be combined to form a xi vector-matrix, where xi = [[xi(N)]] represents the message data point of the ith SU. by $\mathbf{X} = [x1, x2, \ldots, xM]^T$. Each SU calculates its energy level and relays that information to another SU working as a fusion centre for cooperative sensing. All SUs communicate their energy levels to the fusion centre (FC), which calculates the channel availability based on that information. There is a good deal of PUs that are alternately active and inactive in this paper.

Energy Vector Model:

An SU needs to perform energy detection for a period of time equal to to approximate the energy level. For a given period of time, the energy detector records w samples of the complex baseband signal. Assume that SU n collected the ith sample of signals, which is represented by the emblem Zn (i). We used the sum of all active PUs' signals and thermal noise to create the signal samples. $$Z_n(i) = \sum_{m=1}^{M} S_m h_{m,n} X_m(i) + N_n(i) \qquad (1)$$

Here, $h_{m,n}$ represents the PU m to SU n channel gain, and the signal from PU m is $X_m(i)$. The thermal noise at SU n is $N_n(i)$.

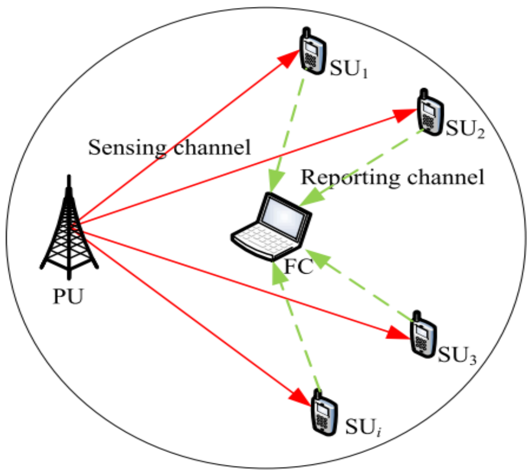


Figure 1: A high-level overview of the E^2SS framework

An energy level standardized by noise frequency response can be measured, the detector of SU n utilizes signal samples (2)

$$X_n = \frac{2}{\eta} \sum_{i=1}^{w\tau} \left| Z_m(i) \right|^2 \qquad (2)$$

This "energy vector" is generated by the fusion centre from all of the SUs reporting their estimated energy levels.

$$X = \left(X_1, \ldots, X_N \right)^T \qquad (3)$$

A. Operation of Proposed E^2SS Framework:

We can accurately estimate the channel availability A from an energy vector X using an energy-efficient spectrum sensing (E^2SS) technique. To build a classifier in machine learning terms, creating an energy vector X that correctly maps to channel availability A is the same. The word "energy vector" in our trouble match up to the machine learning term "feature." First, collect as many training energy

International Journal of Global Science Research
Vol. 12, Issue. 2, October 2025, pp. 2631-2644
Available Online at www.ijgsr.com

ISSN: 2348-8344 (Online)
DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305

vectors as feasible for the classifier.

We can denote the training energy vectors by their respective numbers, and we can denote the channel availability by their respective numbers, x(l) (l). This sequence instructs the classifier to learn the set of having trained energy vectors, which are represented by variables x=x(1),..., x(L) (where L represents the amount of training data), by passing them to the classification model as input for training. It is unnecessary to label every training energy vector with the proper channel availability for unsupervised learning. As an alternative to unsupervised learning, supervised learning requires a set of channel availabilities, denoted by the notation of the type of the set of training energy vectors of the form of $a = \{a(1), . . . ,a(L)\}$. We followed this by training the classifier and refining it by means of the preparation power vectors. Depending on the machine learning method, the training process will vary.

It's time to put the classifier to the test. Assume $x*$ be the test energy vector that it has received and assume $a*$ represent associated channels availability. Consider a~ also represent the classifier's determination of the channel's availability. It divides the energy vector x* into two classes: "channel available class" and "channel unavailable class", depending on if the channel is available. There is no PU active, so the CR network can access the channel, and the electricity vector is in the class of channels available. Because of this, the availability of the channel is properly identified when condition a~ = $a*$ is met, but misdetection takes place when a~ is 1, and $a*$ is -1.

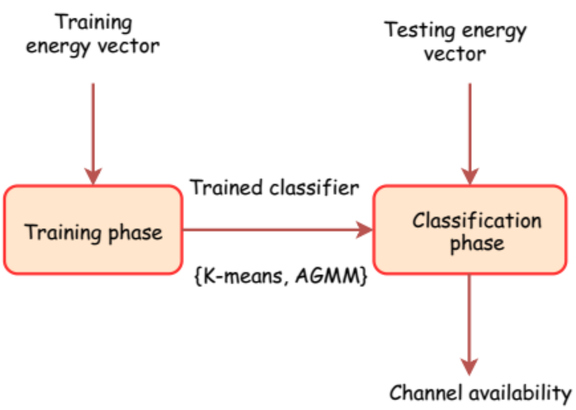


Figure 2: The E²SS framework modular structure

Figure 2 depicts the proposed framework modular structure, which is divided into two parts: training and classification. The training and classification stages may work independently under this design. A test energy vector is generated and sent to a classification module whenever the channel availability has to be determined by the CR network. Using the a trial vector for determining energy usage & classifier, we were able to identify the channel's availability. Finding the availability of a channel in the CR network usually takes just a short time. Because of their low complexity, the suggested methods' classification delay meets this criterion. The training module's purpose is to use the training energy samples to create a trained classifier, passing along to the classification module. When deploying the CR network for the first time or during the change in the radio, the training module may be activated. To stay informed of the ever-changing environment, the CR network may occasionally activate the training module.

B. Investigation of Unsupervised learning models for CSS:

In this section, we used particular relevance to the cooperative spectrum sensing we examine unsupervised learning methods. To train a classifier using unsupervised learning, all that's given to it is the training energy vectors (i.*e.*, $x = \{x(1), . . . , x(L)\}L$)). Because every learning energy vector is analogous to the channel availability information

$a = \{a(1), . . . , a(L)\}$ is not required for unsupervised learning methods, they may use the information in the training energy vectors themselves. We'll now look at how well two typical unsupervised clustering algorithms, K-means clustering and the AGMM, relate to CSS. Each time the classifier is presented with a test ascertain for classification, the classifier is trained using these clustering algorithms. To determine whether or not a test energy vector is part of an available channel class, the classifier looks for a cluster to which it belongs. When a test electricity vector belongs to the first cluster, the classifier assigns it to the available class.

International Journal of Global Science Research ISSN: 2348-8344 (Online)
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305
Available Online at www.ijgsr.com

K-Means Clustering Algorithm:

To test the **K-Means** we consider a preparation power vectors (e.g., $x = x(1),\ldots,x((L))$ is partitioned into K displace groups using the unsupervised K-means clustering method. Consider cluster k as a collection of training energy vectors and call it Ck. The centroid of cluster k is αk. Clustering using K-means attempts to discover K groups of clusters that minimize the sum of squares inside each cluster $C1,\ldots, CK$:

$$\underset{C_1,\ldots,C_K}{\arg\min} \sum_{k=1}^{K} \sum_{x^{(l)} \in C_K} \left\| x^{(l)} - \alpha_k \right\|^2 \quad (4)$$

We make use of Algorithm 1's incremental suboptimal algorithm to locate clusters that meet equation (4).

Algorithm 1: for CSS K-Means Clustering

$\alpha 1 \leftarrow \mu_{X \mid S=0}$

α_k is initialized $\forall k = 2,\ldots K$

while C_k for some $k$value, tuned in the prior iteration **do**

$$C_k \leftarrow \left\{ \begin{array}{l} x^{(l)} \, \mathrm{P} \, x^{(l)} - \alpha_k \le \left\| x^{(l)} - \alpha_i \right\|, \\ \forall i = 1,\ldots\ldots, K \end{array} \right\}$$

$\forall k = 1,\ldots., K$

$$\alpha_k \leftarrow \left| C_k \right|^{-1} \sum_{x^{(l)} \in c_k} x^{(l)}, \forall k = 2,\ldots, K$$

end while

At the end of the training phase, we provide the classifier with the test energy vector y* for classification. An energy vector can be classified into one of two groups based on its distance from its centroids: Cluster 1 or Cluster 2 based on its distance from the centroids. As long as we meet the following criterion, the channel unavailability class (i.e., a $\tilde{}$ = −1) is classified by the classifier y* .

$$\frac{\left\| y^{*} - \alpha 1^{*} \right\|}{\min k{=}1,\ldots,K \left\| y^{*} - \alpha 1^{*} \right\|} \ge \beta \quad (5)$$

It is possible to classify y (i.e., the channel-available class) as a = 1 instead. It is the parameter that sets the limit at which misdetection and false alarm probabilities are balanced. An increase in the value of increases the probability that we will classify it as the channel available class, improving the probability of misdetection while simultaneously reduces the probability of a false alarm being generated.

Need for Gaussian Mixture Model over K-Means:

Instead of assigning a flag indicating that a point belongs to a certain cluster as in traditional K-Means, give a probability that the point belongs to that cluster to each individual. A non-convex cluster is generated using GMM, and the variance of the distribution may be used to regulate it. Cluster covariance can be handled much more creatively with GMM than with other models. K-means is a variant of the general mixed model in which the covariance of each cluster in all dimensions is close to zero. According to this, a point will only be allocated to the cluster to which it is the closest. The unconstrained covariance structure of each cluster may be achieved via GMM. Instead of K-means' spherical distribution, consider an extended or rotated one. Cluster assignment in GMM is considerably more flexible than in K-means as a consequence.

Proposed Adaptive Gaussian Mixture Model:

We typically estimated GMM parameters from the training energy vector by means of Expectation-Maximization (EM) technique estimation using a well-trained previous model. We made the following assumptions while considering a mixture model: a specific representative example An energy vector (xi) is collection of numerous different group trained energy vectors (Davanam et. al., 2021).

$$P\left(x_i \mid \theta\right) = \sum_{c=1}^{C} \alpha_c \, \phi\left(x_i \mid \theta\right) \quad (6)$$

which denotes a D-dimensional data stream vector characteristics , Given a data set that contains N models, the probability that the cth categorization will be selected is P will be

International Journal of Global Science Research **ISSN: 2348-8344 (Online)**
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 **DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305**
***Available Online* at www.ijgsr.com**

chosen at random is chosen and $\sum_{c=1}^{C} \alpha_c, \phi(x_i|\theta)$ signifies the The cth Probability distribution model component's Gaussian distribution density and function are expressed as

$$\phi(x_i|\theta) = \frac{1}{\sqrt{2\pi\sigma_c}} \exp\left[-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right] \quad (7)$$

Where the mean value is μ_c, and the deviation is σ_c^2, Complete Gaussian mixture models are standardized using the mean, deviation and mixed weight training of every module; these factors are stated in the equation.

$$\theta = \left\{\mu_c, \delta_c^2, \alpha_c\right\}, (c = 1, ..., C) \quad (8)$$

We used the estimation maximization algorithm to actually the GMM parameters time after time training dataset and the GMM configuration. We closely distributed the trained eigenvectors as determined by the expectation maximisation process. If we assume each vector is independent of the others and if N training vectors $x_i = [x_{1n}, x_{2n}... x_{in}..., x_D]$ are used, we can state the GMM probability as

$$p(X_i|\theta) = \prod_{n=1}^{N} p(x_n|\theta) \quad (9)$$

Maximizing the GMM likelihood function's logarithm likelihood function is done in the following manner:

$$\max_{\theta} \ln p(X_i|\theta) = \sum_{n=1}^{N} \ln \sum_{c=1}^{c} \alpha_c \phi(x_i|\theta_c) \quad (10)$$

Rather than explicitly optimising parametersθ, iterative maximisation likelihood parameter estimation is used to generate a maximized expectation in a particular situation for the non-linear functions for the parameter discussed above (Chris and Adrian, 2002). Fundamental to the EM algorithm is estimating a novel constraint starting the primary constraint, resulting in the $p(X_i|\bar{\theta}) \geq p(X_i|\theta)$.This process continues until we reach a particular convergence threshold, at which point we utilize the new parameter as the initial point for the following iteration. In the clustering process, we predetermined the number of clusters C in advance. The variable $\Pr(c|x_n, \theta)$ added shows the likelihood of nth instruction information from the cth form being used in the final model. The following is the formula for calculating the posterior probability:

$$\Pr(c|x_n, \theta) = \frac{\alpha_c \phi(x_i|\theta_c)}{\sum_{n=1}^{C} \alpha_c \phi(x_i|\theta_c)} \quad (11)$$

Methods seen below are used to ensure that parameter likelihood increases monotonically in each EM iteration.

$$\bar{\alpha}_c = \frac{1}{N} \sum_{n=1}^{N} \Pr(c|x_n, \theta) \quad (12)$$

$$\bar{\mu}_c = \frac{\sum_{n=1}^{N} \Pr(c|x_n, \theta) x_n}{\sum_{n=1}^{N} \Pr(c|x_n, \theta)} \quad (13)$$

$$\bar{\sigma}_c^2 = \frac{\sum_{n=1}^{N} \Pr(c|x_n, \theta) x_n^2}{\sum_{n=1}^{N} \Pr(c|x_n, \theta)} - \bar{\mu}_c^2 \quad (14)$$

where, μ_c, $\sigma_c^2$and x_n refer to some of the essentials μ_c, σ_c^2& x_n, correspondingly.

Once the optimum parameter θ has been determined, the classifier gets the test energy vector x∗ for classification and uses it to determine the next optimal parameter. Using the test energy vector x∗, the classifier evaluates whether it belongs to cluster 1. We assign the unavailable channel class to x∗ (i.e., $a\sim = -1$) only if for a specified threshold δ. With an increase in, we may reduce false alarm probabilities while also improving misdetection probabilities δ because the channel accessible class of $x*$ is more likely to be identified.

The AGMM classifier's role is to assign various clusters to the input energy vectors based on their features. It's impossible to identify the training energy vectors of an AGMM unsupervised classifier to show whether they belong to a particular cluster. The AGMM classifier uses a linear mixture of multi -variate Gauss probabilities (pdfs) to model the

International Journal of Global Science Research **ISSN: 2348-8344 (Online)**
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 **DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305**
***Available Online* at www.ijgsr.com**

observation vector's conditional pdf. Each of the aspects like this:

$$f\left(x_i,\mu_c,\sigma_c\right)=\frac{1}{\sqrt{2\pi\sigma_c}}e^{\frac{-\left(x_i-\mu_c\right)^2}{2\sigma_c{}^2}} \qquad (15)$$

In eq. (8), Each feature's standard deviation are contained in the d-component feature vector, which is xi. In addition, the features are assumed to be independent of one another.To put it another way, $P(x_i)$ may be expressed as a product of multivariate probability densities for every elements of x_i., AGMM is a modified version of GMM with two extra parameters, n and N. Also, N stand for the entire energy test vectors in the data. N stand for the quantity of samples in every cluster. σ is identical to GMM in that the probability density is a function of x_i, μ_c.

$$f\left(x_i,\mu_c,\sigma_c,\pi,N\right)=\frac{1}{\sqrt{2\pi\sigma_c}}\left(\frac{\mu_c}{N}\left(e^{\frac{-\left(x_i-\mu_c\right)^{n+1}}{2\sigma_c{}^2}}+\frac{\mu_c}{N}\sum_{i=1}^{n}\left(\frac{x_i-\mu_c}{\sigma_c}\right)^{\frac{\mu_c}{\sigma_c}}\right)\right) \qquad (16)$$

Each sample $x_i$is a D-dimensional vector, which is the probability density function of the Adaptive GMM. Mean, and standard deviation are computed separately since the features are distinct. The mean μ_c and standard deviation of each feature x_i in a cluster of n samples are computed by adding the x_i values from all the samples in that cluster together. Equations 10 and 11 provide the mean and standard deviation, respectively.

$$\mu_c=\sum_{j=1}^{n}\frac{x_{ij}}{n} \qquad (17)$$

$$\sigma_c=\sqrt{\sum_{j=1}^{n}\frac{\left(x_i-\mu_c\right)^2}{n-1}} \qquad (18)$$

In order to obtain the weighted variance, we weigh the samples according to their proportional share in the overall Gaussian distribution of the data. Until the algorithm reaches a local optimal or reaches its maximum number of iterations, the input variables of the Gaussian kernel are kept constant and updated. The AGMM-based clustering process is fully described in Algorithm 2.

Algorithm 2: The AGMM is used to group data.

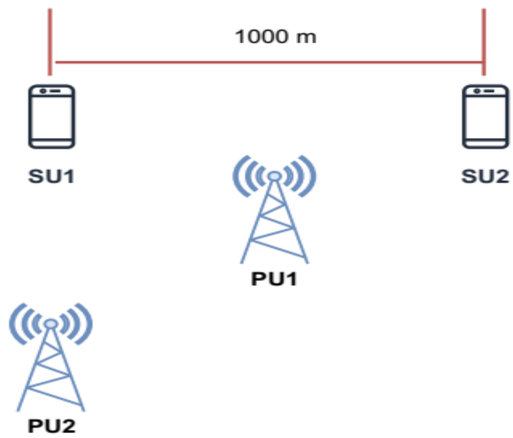
Input: x_i = [x_{1n}, x_{2n}... x_{in}…, x_D], Cluster number C, Sample number N

Output: separate X_i into c clusters {C_1, C_2,….,C_m}, where $\bigcup_{i=1}^{c} C_i = X_i$

1. Initialization of the parameters of Gaussian model $\theta=\left\{\mu_c,\sigma_c^2,\alpha_c\right\}\left(c=1,....,C\right)$
2. repeat
3. **for n ← 1 to N do**
4. **forc ← 1 to C do**
5. Compute the probability of nth training data x_n from the c model
6. **end for**
7. **end for**
8. **forc ← 1 to C do**
9. Computing the parameters of the cth Gaussian model $\mu_c,\sigma_c^2,\alpha_c$
10. Update the parameters $\mu_c,\sigma_c^2,\alpha_c$
11. **end for**
12. **until** the termination condition is satisfied

Analysis of two cases of user locations:

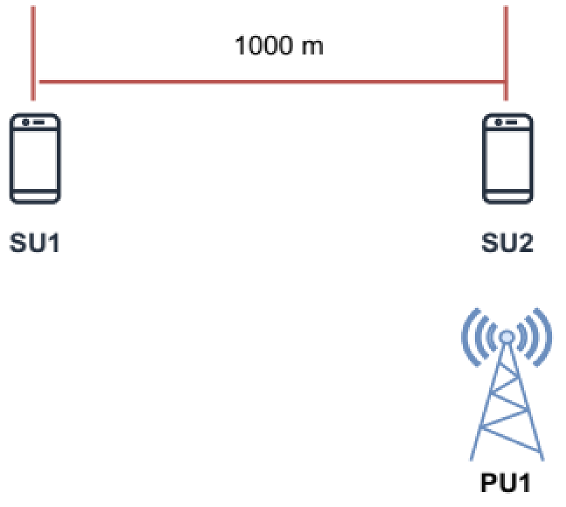
As a way to demonstrate the potential benefits of CSS techniques, we present two examples of energy vectors from two SUs in two scenarios. The locations of the two PUs in Case 1 are depicted in Figure 3(a).



(a)

International Journal of Global Science Research
Vol. 12, Issue. 2, October 2025, pp. 2631-2644
Available Online at www.ijgsr.com

ISSN: 2348-8344 (Online)
DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305



(b)

Figure 3: Locations of users in two different scenarios (a) PU and SU locations in case 1 (b) PU and SU locations in case 2

In case 1, the PUs are triggered based on the likelihood of v((0, 0)T) = 0.55, v((0, 1)T) = 0.32, v((1, 0)T) = 0.2428, v((1, 1)T) = 0.18, and v((1, 1)T)= 0.18, respectively. The PU in Case 2 is the only one, and we depict its location in Figure 3(b). PU is triggered in this case with a probability of v(1) = 0.4, which corresponds to the probability of PU triggering.

Results:

This section includes the results of a simulation study we conducted to assess the proposed scheme's performance. There are 25 SUs in total spread across a 4000m by 4000m area, based on our calculations. This grid topology is used for cooperative spectrum sensing (CSS). Table 1 shows the values of critical simulation parameters. Each PU has a transmission power of 250 mW and 300 mW. Assume two PUs, each with a predetermined location at (1000 m, 1000 m) and (−2000, 0 m), respectively. There is a 0.5 percent chance that a PU will be in the effective state at any given time. Uses MATLAB (R2016a) to enforce the algorithms in 64-bit computers with an Intel Core I5 processor and 8GB of RAM.

Table 1: Network simulation parameters

Parameter name	Value
Bandwidth$w$	5MHz
Sensing duration τ	100μs
Noise spectral density η	-174 dBm
Path-loss exponent α	4

Results and Analysis of proposed and traditional machine learning-based CSS framework:

We show two SU energy vectors in two distinct situations as scatter plots in figures 4-7 to show CSS performs well. The suggested technique's energy vectors and decision surfaces are shown in these plots for various scenarios. The surface splits the energy vectors into two decision regions to decide whether channels are available or unavailable. We showed the decision surface using the Adaptive Gaussian mixture model (AGMM) approach; one suggested method. Figures 4-7 have a classification threshold of 0 in the AGMM (i.e., δ). This is case 1's plots (Figures 4 and 5), whereas case 2's plots (Figures 6 and 7) are drawn. Figures 4 and 5 show a transmission power of 300 mW, whereas figures 6 and 7 show 250 mW.

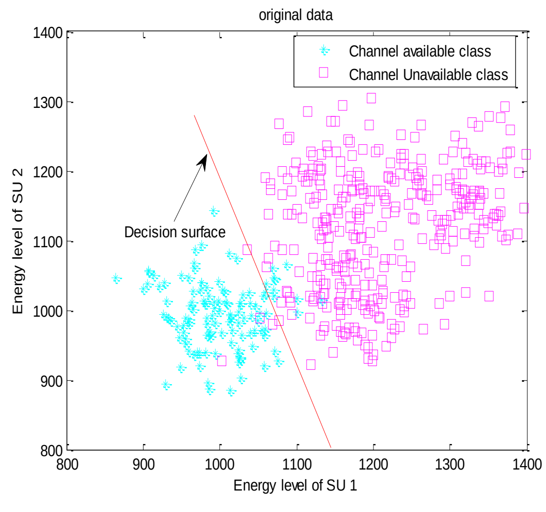


Figure 4: Graph illustrating the distribution of energy vectors in case 1 where the energy transmitted of each primary user is 300 mW for clustered data data

International Journal of Global Science Research
Vol. 12, Issue. 2, October 2025, pp. 2631-2644
Available Online at www.ijgsr.com

ISSN: 2348-8344 (Online)
DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305

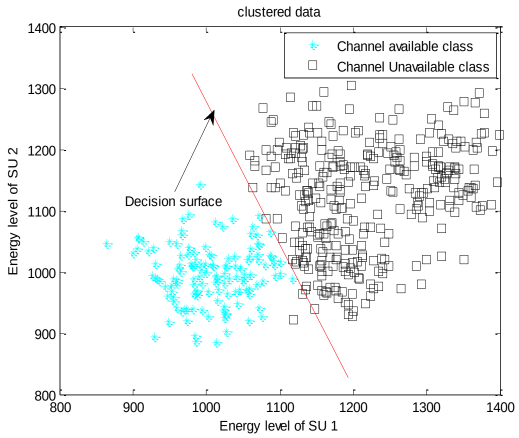


Figure 5: Graph illustrating the distribution of energy vectors in case 1 where the transmit power of each primary user is 250 mW for clustered data data

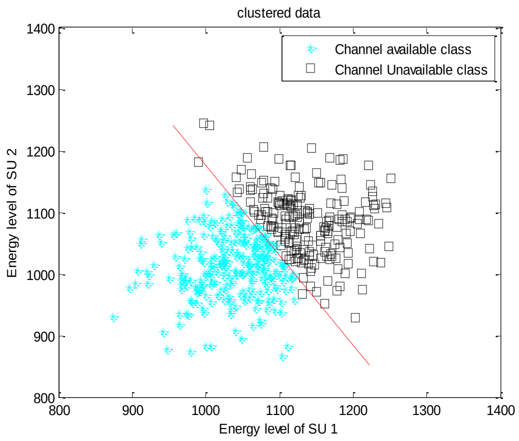


Figure 6: Graph illustrating the distribution of energy vectors in case 2 where the transmit power of each primary user is 250 mW for clustered data

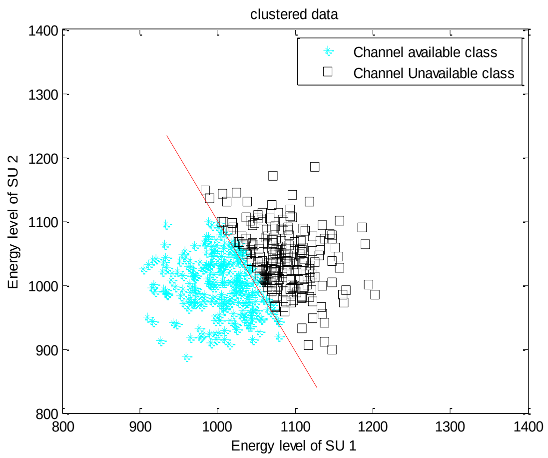


Figure 7: Graph illustrating the distribution of energy vectors in case 2 where the transmit power of each primary user is 250 mW for clustered data

AGMM can adapt to a variety of scenarios, as shown above. Assume that we configure the CR network in case 1 and that each PU's transmission power is 300 mW. Figure 4 depicts the decision surface of the CR network in this scenario. If this is the case, the PU network's configuration is now to case 2. After gathering energy vectors, the CR network may adjust to this change and repeat the decision surface calculation as illustrated in Figure 6. This procedure runs on its own and doesn't need any help from the user.

Table 2: A 300 mW communication energy comparison between two classifiers shows which classifier performs better.

Classifier name	Accuracy in %	Recall in %	F1-score in %
K-means	96.67	99.13	91.05
AGMM (Our work)	99.38	99.84	98.77

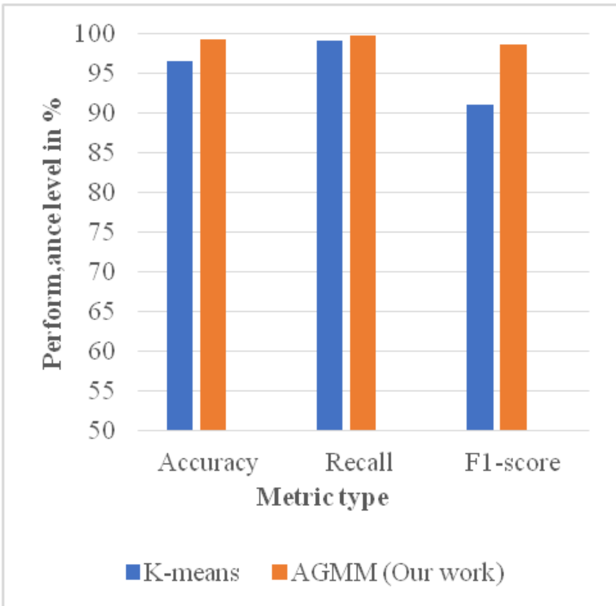


Figure 8: Performance comparison of two classifiers for PU transmit power is at 300mW

International Journal of Global Science Research
Vol. 12, Issue. 2, October 2025, pp. 2631-2644
Available Online at www.ijgsr.com

ISSN: 2348-8344 (Online)
DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305

Figure 8 shows that the proposed AGMMM is 99.38% more accurate than conventional K-means when the PU transmit power is 300mW. AGMM outperformed the traditional K-means classifier, where recall was 99.84%, and F1-score was 98.77%, proving that the proposed AGMM is adaptable to changing environments without requiring further training.

Table 3: Performance comparison of two classifiers for transmit control of every PU is 250 mW

Classifier name	Accuracy in %	Recall in %	F1-score in %
K-means	96.46	99.71	93.33
AGMM (Our work)	97.92	99.72	95.97

Figure 9 shows that the proposed AGMMM has a better accuracy of 97.92% when the PU transmit power is set to 250 mW than conventional K-means. The AGMM classifier also outperformed the traditional K-means classifier, where recall was 99.72%, and F1-score was 95.97%, because of the flexibility of the proposed AGMM to the changing environment without having to train it all over again.

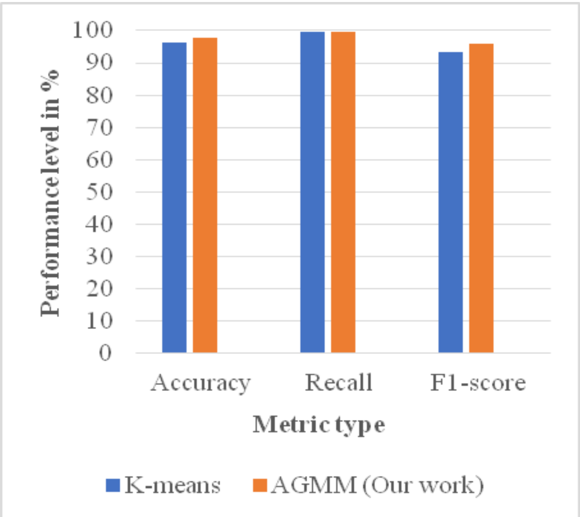


Figure 9: Performance comparison of two classifiers for PU transmit power is at 250mW

Table 3: Performance comparison of two classifiers for classification error

Classifier name	Classification errorfor 300 mW	Classification errorfor 250 mW
KNN	3.33	3.54
AGMM (Our work)	0.62	2.08

For PU transmit power of 250mW and 300mW, we show the classification error of two classifiers in Figure 10. Compared to conventional K-means, the suggested AGMM has low error rates of 0.62 for 300mW and 2.08 for 250mW. For example, training the AGMM cognitive classifier takes just a short time. As a result, this classifier is suitable for CSS, requiring constantly updating the training energy vectors.

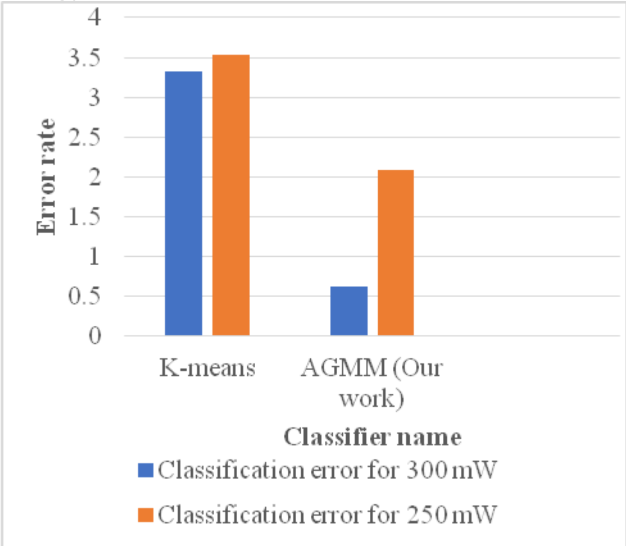


Figure 10: Classification error of two classifiers in case ofPU transmit power is at 250mW and 300mW

For multiple positions of assists SUs, only one PU in Figure 11 compares the receiver operating characteristic (ROC) curves suggested and conventional CSS methods (500 m, 500 m). These results reveal that the proposed AGMM classifier's performance improves as the number of SUs increases. The suggested AGMM method outperforms the currently used K-means

International Journal of Global Science Research
Vol. 12, Issue. 2, October 2025, pp. 2631-2644
Available Online at www.ijgsr.com

ISSN: 2348-8344 (Online)
DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305

technique, which is critical to keep in consideration. It displays the ROC curves when the AGMM Classifier reaches a high detection probability at pf = 0.7 with 3 × 3 SUs (i.e., a total of 9 SUs).

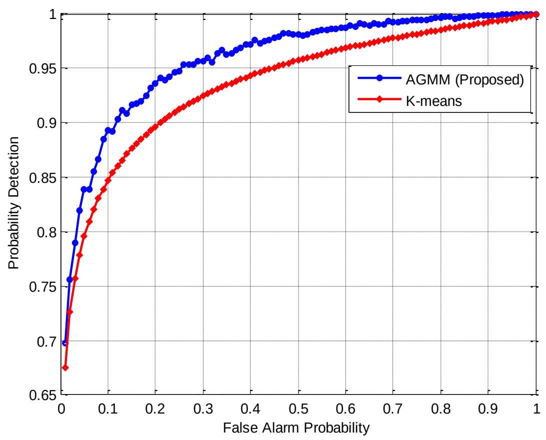


Figure 11: ROC curve for 3-by-3 SU cooperation.

As shown in Figure 12, when just a single PU (500 m, 500 m) is available, the suggested CSS schemes perform better than conventional CSS schemes as measured by receiver operating characteristic (ROC) curves. These results reveal that the proposed AGMM classifier's performance improves as the number of SUs increases. The suggested AGMM method outperforms the currently used K-means technique, which should be noted. It displays the ROC curves when there are 5 × 5 SUs in total (i.e., 25 SUs). The AGMM Classifier has a good detection probability even when low p_f =0.4 is used. To train each classifier, we use 500 training energy vectors.

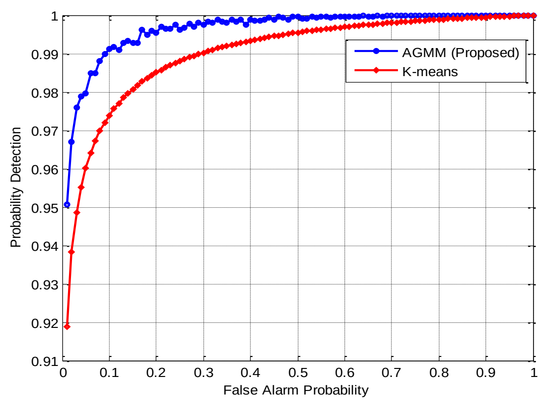


Figure 12: ROC curvesfor 5-by-5 SU cooperation

Figure 13 depicts the suggested scheme's detection performance in a fading environment. Because of fading, we need more energy samples to determine accurately whether a PU is on or off. Magnitude gain is constant in non-fading environments, but it fluctuates in fading environments. As a result, detection probability is proportional to the instantaneous signal-to-noise ratio (SNR).

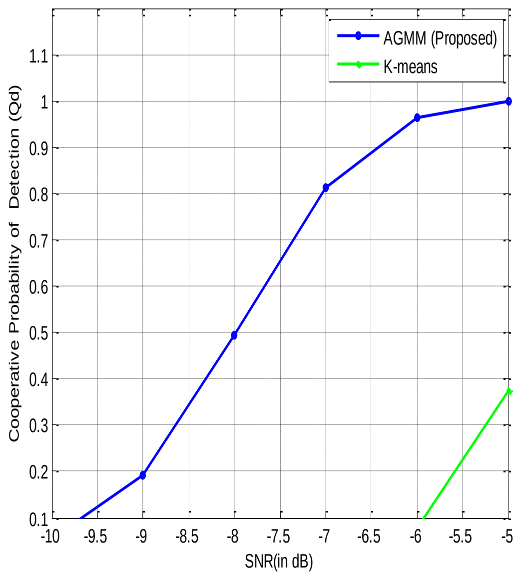


Figure 13: System detection performance with fading channels

AGMM with a more extended training phase outperforms K-means by 15% when SNR is -9 dB, but this outperforms K-means by approximately 35% when SNR improves to -6dB. When SNR circumstances improve, the suggested method beats K-means in terms of discovery routine. Using the projected method in a departure setting, Figure 14 illustrates how well it performs in terms of errors. Noise reduction reduces the inaccuracy rises. The error likelihood is slightly over one with a sensitivity of -10dB. The K-means error probability is 1.1 because of fading, making it much higher than our suggested solution. Graph illustrating the distribution of energy vectors in case 2, where the transmit power of each primary user is 250 mW for clustered data.

International Journal of Global Science Research **ISSN: 2348-8344 (Online)**
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 **DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305**
***Available Online* at www.ijgsr.com**

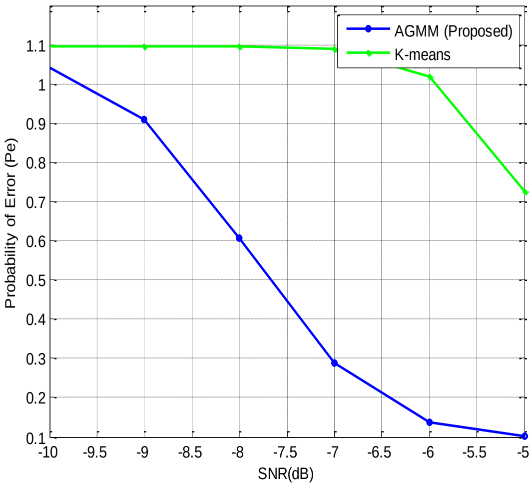


Figure 14: System error performance with fading channels

Conclusion:

We base the reliable spectrum sensing method presented in this paper on machine learning. Unsupervised learning has been used to create Networks based on cooperative spectrum sensing for cognitive radio (CSS). Our proposed unsupervised classifiers for CSS include K-means grouping and the Responsive Gaussian mixture design .Secondary users' (SUs) received we properly considered energy levels while evaluating channel availability. F1 score and classification error and ROC curve measurements are used to assess classifier performance. Training the classifier incorrectly will lead to incorrect decisions. Thus accuracy is critical in a real-world application. Classifiers can be trained more effectively using the energy vectors obtained one at a time by modifying the CSS methods already suggested. Adapting to a changing environment does not need retraining the classifiers. According to simulation results, our suggested method has higher detection performance and better utilization of spectral holes than traditional K-means.

References:

Mitola, J. (2000) Cognitive Radio-An Integrated Agent Architecturefor Software Defined Radio (Ph.D. thesis), KTH, Stockholm,Sweden.

Kolodzy, P. (2002) “Spectrum policy task force,” Rep. ET Docket 02-135, Federal Communications Commission, Washington, DC, USA.

Akyildiz, I. F., Lo, B. F. and Balakrishnan, R. (2011) “Cooperative spectrum sensing in cognitive radio networks: A survey,” Physical Communications, vol. 4, no. 1, pp. 40–62..

Khatereh, A. and Jamshid, A. (2018) Signal Classification for Detecting Primary User Emulation Attack in Centralized Cognitive Radio Networks. 342-347. 10.1109/ICEE.2018.8472515.

Lakshmanan, T. and Hyils, M. (2020) A novel system model for detecting primary user emulation attack in cognitive radio networks. International Journal of Wireless and Mobile Computing. 18. 381. 10.1504/IJWMC.2020.108538.

Li, Yongcheng; Ma, Xiangrong; Wang, Manxi; Chen, Huifang and Xie, Lei. (2019) Detecting Primary User Emulation Attack Based on Multipath Delay in Cognitive Radio Network. 10.1007/978-981-10-8968-8_31.

Park, Jongbu and Lim, Chang, (2020) Multiple Detections Based Spectrum Sensing Method for Radar Bands. The Journal of Korean Institute of Electromagnetic Engineering and Science. 31. 925-932. 10.5515/KJKIEES.2020.31.11.925.

Jayanthi, V. E., Mohan, K., Sriram, S., Jagannath, M., Harshavarthanan, D. and Adalarasu, K. (2020) Power Spectral Density for Detecting White Space in Wavelet Based Spectrum. International Journal of Advanced Science and Technology. 29. 7148-7153.

Z. Han, R. Zheng, and H. V. Poor, (2011) “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.

Ganesh, Davanam, T. Pavan Kumar, and M. Sunil Kumar (2020) "A Dynamic and adaptive learning mechanism to reduce cross layer attacks in cogntive networks." *Materials Today: Proceedings*.

M. Bkassiny, S. K. Jayaweera, and K. A. Avery (2011) “Distributed Reinforcement Learning based MAC protocols for autonomous cognitive secondary users,” in Proceedings of the 20th Annual Wireless and Optical Communications Conference, (WOCC 11), Taiwan.

International Journal of Global Science Research **ISSN: 2348-8344 (Online)**
Vol. 12, Issue. 2, October 2025, pp. 2631-2644 **DOI: https://doi.org/10.26540/ijgsr.v12.i2.2025.305**
***Available Online* at www.ijgsr.com**

Sangamithra, B., Neelima, P., & Kumar, M. S. (2017, April). A memetic algorithm for multi objective vehicle routing problem with time windows. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)* (pp. 1-8). IEEE.

B. Y. Reddy (2008) "Detecting Primary Signals for Efficient Utilization of Spectrum Using Q-Learning," in Proceedings of the 2008 Fifth International Conference on Information Technology: New Generations (ITNG), pp. 360–365, Las Vegas, Nev, USA.

Q. Zhu, Z. Han, and T. Bas (2010) "No-Regret Learning in Collaborative Spectrum Sensing with Malicious Nodes," in Proceedings of the 2010 IEEE International Conference on Communications, pp. 1–6, Cape Town, South Africa.

Ahmad, Hassaan. (2019) Ensemble Classifier Based Spectrum Sensing in Cognitive Radio Networks. Wireless Communications and Mobile Computing. 2019. 1-16. 10.1155/2019/9250562.

Hassan, Yasmin and El-Tarhuni, Mohamed and Assaleh, Khaled. (2012) Learning-Based Spectrum Sensing for Cognitive Radio Systems. Journal of Computer Networks and Communications. 2012. 10.1155/2012/259824.

Ejaz, Waleed and UlHasan, Najam and Aslam, Saleem and Kim, HyungSeok. (2011) Fuzzy Logic Based Spectrum Sensing for Cognitive Radio Networks. 185-189. 10.1109/NGMAST. 41.

Dasmahapatra, Suddhendu and Patnaik, Sharanya and Sharan, Shivendra and Gupta, Mitali. (2019) Performance Analysis of Prediction Based Spectrum Sensing for Cognitive Radio Networks. 271-274. 10.1109/ICCT46177.2019.8969028.

Karaputugala, Madushan Thilina; Choi, Kae Won; Saquib Nazmus; Hossain, Ekram. (2013) Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks. Selected Areas in Communications, IEEE Journal on. 31. 2209-2221. 10.1109/JSAC.2013.131120.

Chhagan, C. and Pandey, R. (2018) Intelligent selection of threshold in covariance-based spectrum sensing for cognitive radio networks. Wireless Networks. 24. 1-13. 10.1007/s11276-017-1533-y.