Cooperative Spectrum Sensing in Cognitive Radio Networks via an Adaptive Gaussian Mixture Model Based on Machine Learning

Srinivas Samala, Subhashree Mishra, and Sudhansu Sekhar Singh
School of Electronics Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha 751 024, India.
Email: {1981088, subhashree. mishrafet, ssinghfet}@kiit.ac.in

Abstract —Spectrum resources are becoming extremely scarce in modern wireless communication. However, the majority of the currently available spectrum resources have not been fully utilized. To mitigate this problem, we suggested Machine learning-based Adaptive Gaussian Mixture Model (AGMM) for cooperative spectrum sensing in cognitive radio networks for pattern classification. We employ the energy level of secondary users to build a feature vector in the proposed method. The training feature vectors for classification are well defined by a combination of Gaussian density functions that are obtained using the proposed method. The proposed method performance is evaluated in terms of accuracy, recall, F1 score, and Receiver Operating Characteristics (ROC) curves. The performance parameters of the proposed method are compared to the existing K-mean clustering approach. As evidenced by the results, the proposed method performs better than an existing method in all comparison parameters, according to the simulation findings in the MATLAB version.

Index Terms—Cooperative spectrum sensing, adaptive gaussian mixture model, cognitive radio networks

I. Introduction

Cognitive Radio (CR) has been applied to wireless communication systems to alleviate the scarcity of radio spectrum by enhancing spectrum utilization [1]. CR is an intelligent wireless device, it can be used to dynamically sense and alter the operational parameters of a radio environment. One of the most essential aspects of CR is spectrum sensing [2]-[4], which allows secondary users (SUs) to access the primary user's licensed frequency band when it is idle. With limited resources, SUs continuously monitor the condition of the Primary Users (PUs) channel and access opportunistically without interfering with PUs. As a result, the efficiency of spectrum sensing is critical in CR. In cooperative spectrum sensing (CSS) CR devices work together to produce better sensing reliability than individual sensing [5], It helps to mitigate PUs issues related to multi-path fading and shadowing [6]-[9]. Cooperative sensing entails the interchange of sensing data between CR devices and the fusion center for decision-making purposes.

Manuscript received April 22, 2022; revised September 16, 2022.
doi:10.12720/jcm.17.10.812-819

II. Related Work

Researchers have looked at machine learning algorithms [10]–[17] to enable CR users to learn from their surroundings. It becomes a task in spectrum sensing for machine learning algorithms to extract an input pattern's feature vector and then classify it into a hypothesis class that either shows PU activity is absent or present. As K-Nearest Neighbors (KNN) belongs to the family of supervised learning methods, they solve the regression and classification problems, it uses training instances to generate K neighborhood classes. Since this algorithm has low complexity for spectrum sensing, it satisfies the needs of CR users. This paper presents a new CSS scheme that is based on machine learning that uses the Expectation Maximization (EM) approach largely because it allows the dimensionality reduction of data feature during the training of the Adaptive Gaussian Mixture Model (AGMM). In pattern classification, feature vectors are extracted from patterns and then fed to a classifier, based on this classifier assigns the pattern to one of the classes. Specifically, the suggested approach has the benefit of implied having to learn the surroundings in an online setting, including the topology of the PU, cognitive radio networks, and channel fading, among several other aspects.

III. Proposed Model

This study employed a total of N PUs, each of which alternated between inactive and active states. We evaluated a CSS with M SUs, each SU calculating the energy level and transmitting it to another SU that acts as the Fusion Center (FC). After collecting information regarding allowed channels, SUs send it to a Fusion Center (FC) via a channel for reporting. The FC performs a combined processing step and makes decisions [18], [19]. We illustrate the model of a cooperative spectrum sensing system in Fig. 1 (below). The signals gathered by M SUs in a CRN may create a signal vector-matrix $X = [x1, x2, \ldots, xM]^T$. All SUs communicates their energy levels to the fusion center (FC), which calculates the channel availability based on this information. A SU performs energy detection for the period equal to τ, assuming a frequency band of w, the energy detector

collects wτ complex signal samples over the period. Consider the ith signal sample, denoted by the symbol Zn(i), occupied by the nth secondary user. We create signal samples from the sum of the signals from all active PUs and thermal noise. The resultant sampled signal is the summation of all PUs signals in presence of thermal noise is.

$$Z_n(i) = \sum_{m=1}^{N} S_m h_{m,n} X_m(i) + N_n(i) \quad (1)$$

Here, $h_{m,n}$ represents the PU m to SU n channel gain, S_m represents a state of the primary user m (m=1 for active state, m=0 for inactive state), and the signal from PU m is $X_m(i)$, the thermal noise present at SU n is $N_n(i)$.

The estimated energy level at nth SU is with normalized power spectral density is expressed in terms of samples as

$$X_n = \frac{2}{\eta} \sum_{i=1}^{w\tau} |Z_n(i)|^2 \quad (2)$$

The "energy vector" generated by the fusion center can be expressed as.

$$X = (X_1, \ldots, X_M)^T \quad (3)$$

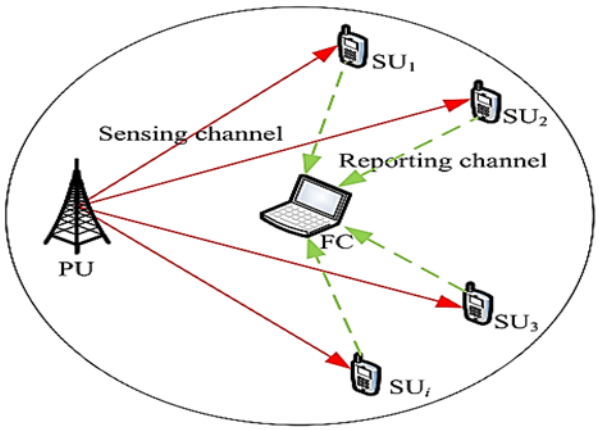


Fig. 1. A high-level overview of the proposed framework.

We can accurately estimate the channel availability A from an energy vector X using the proposed 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 problem corresponds to the machine learning term “feature.” First, collect as many training energy vectors as feasible for the classifier [20]. In this instance, x(l) is the lth learning energy vector, and a(l) is the connection accessibility linked with x(l). The set of learning energy vectors, defined by the variables X = {x (1), . . ., x(L)} in which L signifies the amount of training data is applied to the classifier for training, and 20% of energy vectors are used for testing purpose. Once the training of the classifier is completed then a test energy vector is applied. Assume x∗ be the test energy vector that it has received and assume a∗ represent associated channels availability. Consider a~ represents the classifier’s determination of the channel’s availability. The classifier categorizes the energy vector x* into two classes: “available channel class” (a~ = 1) and “unavailable channel class” (a~ = -1), CR users may access the channel if there is no PU in the active state. Channel is available when the condition a~ = a∗ is met, but misdetection takes place when a~ is 1, and a∗ is -1. Fig. 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 [21], [22]. When deploying the CR network for the first time or during the change in the radio parameters the training module may be activated [23]. To stay informed of the ever-changing environment, CRN may occasionally activate the training module.

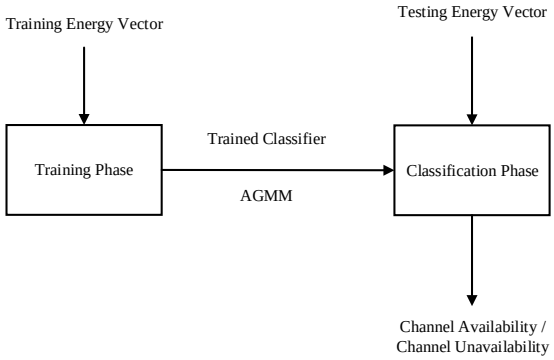


Fig. 2. The CSS framework modular structure

A. *Proposed Adaptive Gaussian Mixture Model for CSS*

To train a classifier using unsupervised learning, first sufficient training energy vectors (i.e., X = {x (1), . . ., x(L)}) are collected, these vectors are used to train the model, and each time it gains a test energy vector for classification. As long as the test energy vector is a member of cluster 1, the classifier considers it to be a member of the channel available class. We typically estimate GMM parameters from the training energy vector using the iterative Expectation-Maximization (EM) technique. We made the following assumptions while considering a mixture model: example x_i represents the awareness of a trained energy vector whose allocation is a combination of many classes trained distributions [24]:

$$P(x_i|\theta) = \sum_{c=1}^{C} \alpha_c \, \phi(x_i|\theta) \quad (4)$$

In Equation (4), the term x_i represents energy vector (i.e. features in classification terminology) for D-dimension, C represents the model’s population and α_c provides c^{th} probability class in the given sample set acquired. Similarly, Equation (5) represents the Gaussian distribution density function that can be rewritten in Equation (6) as

$$\sum_{c=1}^{C} \alpha_c, \phi(x_i \mid \theta) \tag{5}$$

$$\phi(x_i \mid \theta) = \frac{1}{\sqrt{2\pi\sigma_c}} \exp\left[-\frac{(x_i - \mu_c)^2}{2\sigma_c^2} \right] \tag{6}$$

where the mean value is μc, and the deviation is σ^2_c , the equation is a complete Gaussian mixture model that is normalized by taking the average the mean value, variance, and weights of all modeling, and these parameters are stated a

$$\theta = \left\{ \mu_c, \sigma_c^2, \alpha_c \right\}, (c = 1, ..., C) \tag{7}$$

We used the expectation-maximization algorithm to repeatedly estimate the parameters of the GMM. The trained energy vectors are determined by the expectation-maximization 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_{Dn}] are used, we can state the GMM probability as

$$p(X_i \mid \theta) = \prod_{n=1}^{N} p(x_n \mid \theta) \tag{8}$$

Now, use the logarithm of the likelihood function and then maximize it as

$$\max_{\theta} \ln p(X_i \mid \theta) = \sum_{n=1}^{N} \ln \sum_{c=1}^{c} \alpha_c \phi(x_i \mid \theta_c) \tag{9}$$

Rather than explicitly optimizing parameters θ, iterative maximization likelihood parameter estimation is used to generate a maximized expectation in a particular situation for the non-linear functions of the parameter discussed above [25]. In the EM method condition used is

$$p\left(X_i \mid \bar{\theta}\right) \geq p(X_i \mid \theta) \tag{10}$$

This process continues until we reach a particular convergence threshold, at this point, we use the new parameter as the starting values. In the clustering procedure, we predetermined the number of clusters C in advance. The variable $\Pr\left(c \mid x_n, \theta\right)$ added shows the likelihood of the n^{th} preparation data from the c^{th} replica being used in the final model. The following is the formula for calculating the posterior probability:

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

To guarantee the likelihood value of the parameter increases monotonously, for each EM iteration, apply the re-estimation methods as shown in equations (12), (13), and (14)

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

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

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

where, μc, σ^2_c, and x_n denote its elements separately. 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~ = −1) only if for a specified threshold δ. With an increase in δ, we may reduce false alarm probabilities while also improving misdetection probabilities. The AGMM classifier's role is to assign input energy vectors to various clusters based on their features.

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

In Equation (15), x_i signifies the d-component feature vector, μc and σ_c are the d-component vectors containing the mean and standard deviation of each feature, respectively. In addition, we assume the features 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 element of x_i, AGMM is a modified version of GMM with two extra parameters, n, and N. Also, N represents the total number of energy test vectors in the data. N represents the number of samples in each cluster. σ_c is identical to GMM in that the probability density is a function of xi, μc.

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

Each sample x_i is a D-dimensional vector, which is the probability density function of the Adaptive GMM. We compute separately mean and standard deviation since the features are distinct. The mean μc 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 and N represents the total number of

energy test vectors in the data. Equations (17) and (18) provide the mean and standard deviation, respectively.

$$\mu_C = \sum_{i=1}^{n} \frac{x_i}{n} \tag{17}$$

$$\sigma_C = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \mu_C)^2}{n-1}} \tag{18}$$

The weighted standard deviation, equivalent mean is computed by dividing the proportion from every sample to the Gaussian distribution by the number of samples in the group. Thereafter, until the approach converges to the local optimal or until we reached the maximum number of iterations, the original parameters of the Gaussian distribution are updated. Based on the AGMM, Algorithm 1 gives a detailed description of the clustering procedure.

Algorithm 1: AGMM based Clustering operation
Input: xi = [x1n, x2n... xin…, xDn], Sample number N and Cluster number C,
Output: Segment Xi into c group {C1, C2..., Cm}, where $\bigcup_{i=1}^{c} Ci = X\,i$
1. Initialization of the parameters of the Gaussian model $\theta = \{\mu_C, \sigma_C^2, \alpha_C\}(c = 1,....,C)$
2. repeat
3. **for n ← 1 to N do**
4. **for c ← 1 to C do**
5. Compute the probability of nth training data xn from the c model
6. **end for**
7. **end for**
8. **for c ← 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

B. *Analysis of Two Cases of User Locations*

To show the usefulness of the CSS methodologies we used two evidence of energy vector scatter plots from both SUs in two different scenarios. The PUs in case 1 are located in two different locations, which are represented in Fig. 3 (a) and in case 2 PU is located in one location only as shown in Fig. 3(b).

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

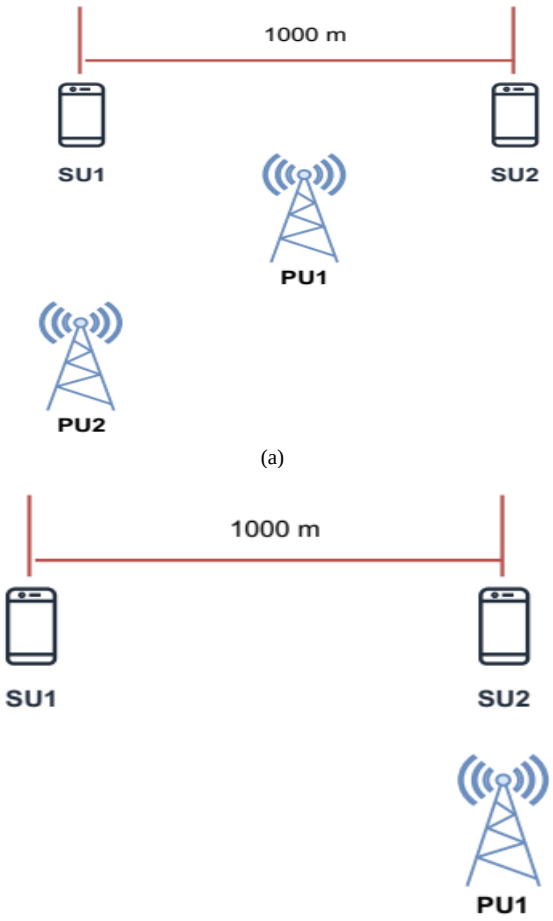


Fig. 3. Locations of users in two different scenarios (a) PU and SU locations in case 1 (b) PU and SU locations in case 2

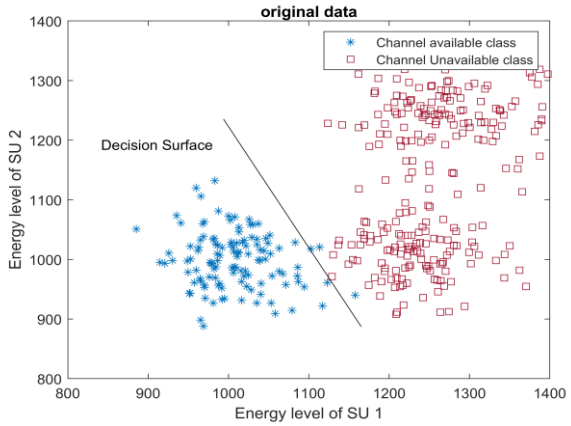
IV. Results and Discussion

This section includes the results of a simulation study and assesses the proposed scheme's performance. According to our calculations, we considered the secondary users to be 5×5 total of 25 SUs with a network size of 4000m × 4000m area. Here, Table I 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 (−1500, 0 m) and, (500 m, 500 m) respectively. The probability of PU's active state is 0.5, and each PU's state is independent of the other PU's. MATLAB (R2016a) is used on a 64-bit computer with a Core i5 (clock speed:2.8 GHz) and 8 GB RAM to implement the algorithms.

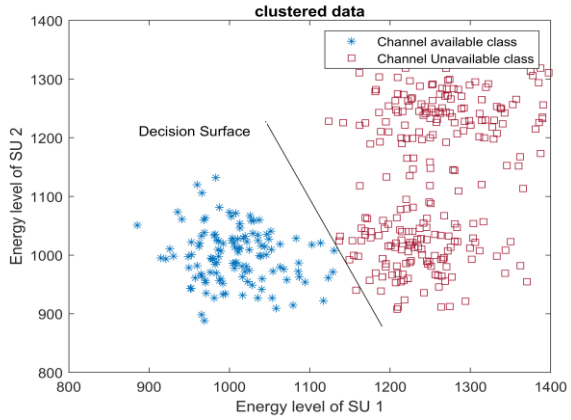
TABLE I. NETWORK SIMULATION PARAMETERS

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

We illustrated two SUs energy vectors in two distinct situations as scatter plots in Fig. 4-5 to show CSS performs well. 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. Fig. 4-5 have a classification threshold of 0 in the AGMM (i.e., δ). Case 1 uses transmission power of 300 mW, plots are shown in (Fig. 4 (a) and (b)), whereas case 2 uses a transmission power of 250mw, plots are shown in (Fig. 5 (a) and (b)).

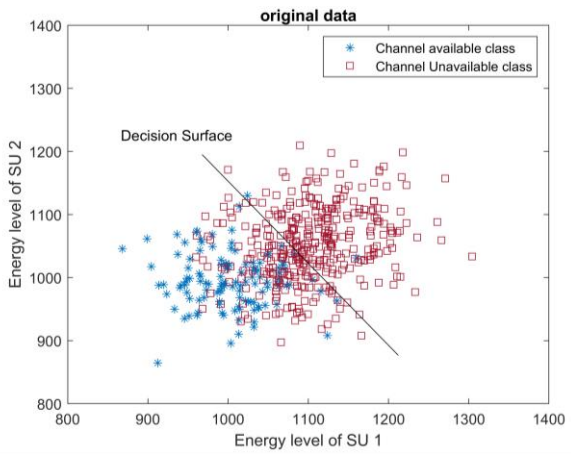


(a)

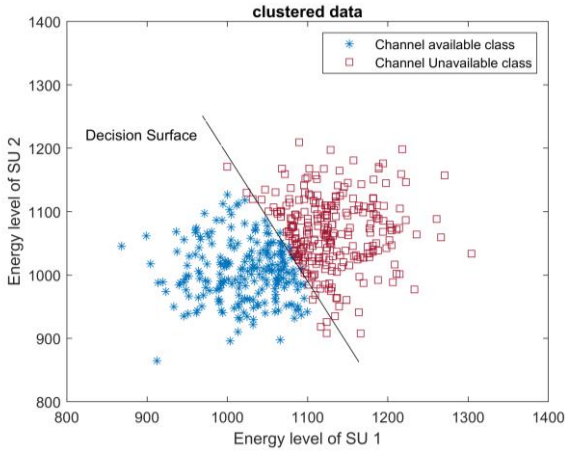


(b)

Fig. 4. The distribution of energy vectors in case 1 when the transmission power is 300 mW (a) for original data. (b) for clustered data.



(a)



(b)

Fig. 5. The distribution of energy vectors in case 2 when the transmission power is 250 mW (a) for original data. (b) for clustered data.

As listed in Table II, we observed the power for every PU is 300mW in this case and applied two classifiers apply to see the performance.

TABLE II. PERFORMANCE COMPARISON CLASSIFIERS WHEN TRANSMIT POWER IS AT 300MW

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

TABLE III. PERFORMANCE COMPARISON CLASSIFIERS WHEN TRANSMIT POWER IS AT 250MW

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

Table II shows that the proposed AGMM is 99.38% more accurate than conventional K-means when the PU transmits 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 III shows that the proposed AGMM 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.

For PU transmission power of 250mW and 300mW, we show the classification error of two classifiers in Table IV. 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 classifier takes just a short time. As a result, this classifier is suitable for CSS, requiring constantly updating the training energy vectors.

TABLE IV. PERFORMANCE COMPARISON TWO CLASSIFIERS FOR CLASSIFICATION ERROR

Classifier name	Classification error for 300mW	Classification error for 250mW
K-means	3.33	3.54
AGMM	0.62	2.08

As shown in Fig. 6(a) compares the receiver operating characteristic (ROC) curves for proposed and k-means methods (500 m, 500 m). These results reveal AGMM classifier's performance improves as the number of SUs increases. The proposed AGMM method outperforms the K-means technique, It displays the ROC curves when the AGMM Classifier reaches a high detection probability at pf = 0.7 with 3 × 3 SUs (i.e., 9 SUs).

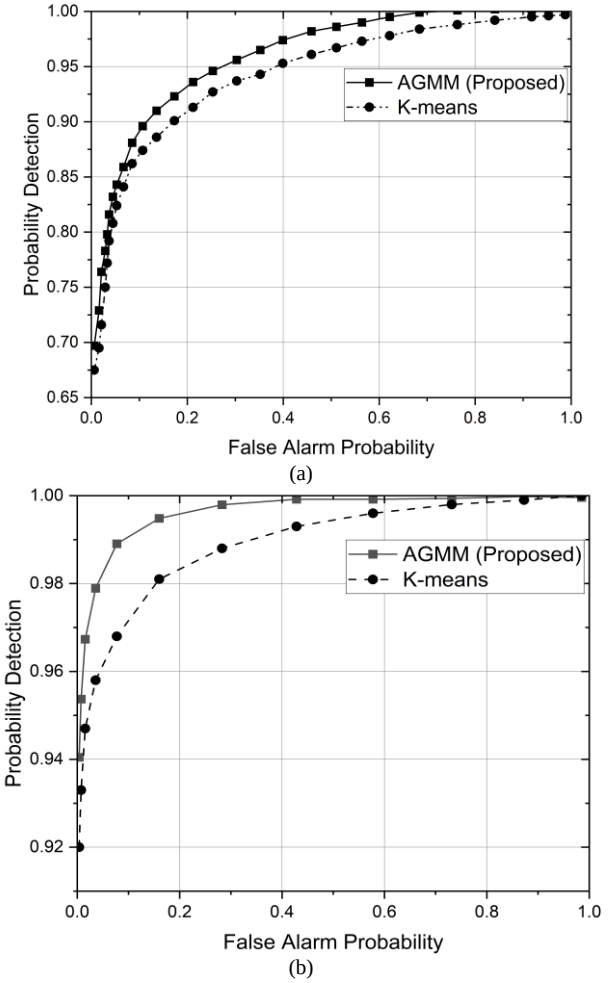


Fig. 6. (a) ROC curve for 3×3 secondary users, (b) ROC curve for 5×5 secondary users

As shown in Fig. 6(b), when just a single PU (500 m, 500 m) is available, the suggested CSS schemes perform better than the k-means, as measured by receiver operating characteristic (ROC) curves. 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 pf =0.4 is used. To train each classifier, we use 500 training energy vectors.

In Fig. 7(a) AGMM with a more extended training phase outperforms K-means by 15% when SNR is -9 dB, outperforms K-means by approximately 35% when SNR improves to -6dB. When SNR circumstances improve, the suggested method beats K-means in terms of detection performance.

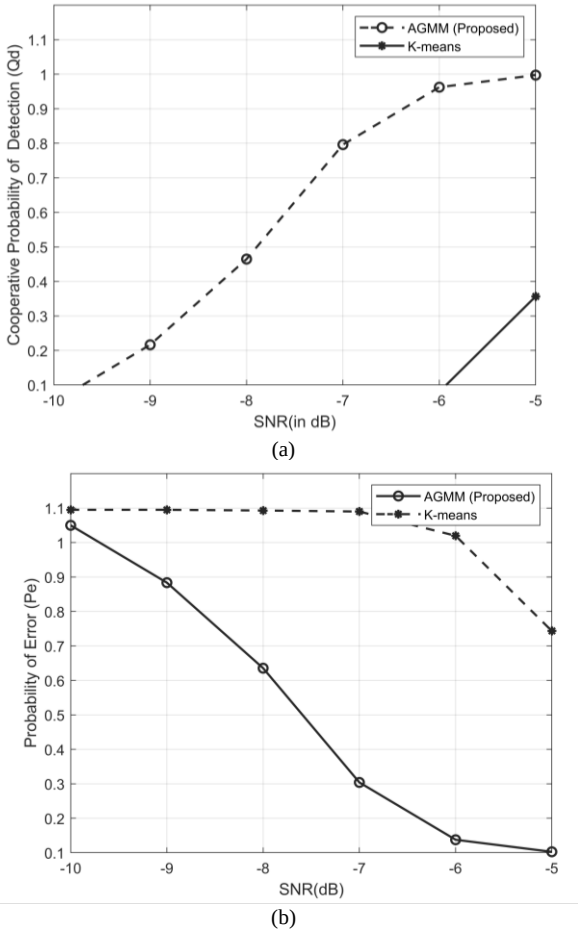


Fig. 7. (a) System detection performance with fading channels, (b) System error performance with fading channels

Using the proposed method in a fading environment, Fig. 7(b) illustrates how well it performs in terms of errors. The inaccuracy decreases as the signal-to-noise ratio (SNR) rises. The error probability 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.

V. CONCLUSION

In this paper, a machine learning-based reliable spectrum sensing scheme is proposed. We have used an unsupervised classifier Adaptive Gaussian mixture model (AGMM) for CSS. In this channel, availability is determined by using energy vectors measured at the SUs. We estimate the performance of the classifier in terms of accuracy, recall, F1-score, classification error, and the ROC curves. To get accurate decisions we need an adequate number of training vectors. The proposed approach can be further boosted by gradually training the

classifier with training vectors obtained one by one. This permits the classifier to adapt to the varying conditions without training all over again. Simulation results show that our proposed scheme has better detection performance and better spectral hole exploitation capability than the conventional K-means.

Conflict of Interest

The authors declare no conflicts of interest.

Author Contributions

Under the supervision of Subhashree Mishra and Sudhansu Sekar Singh, Srinivas samala has examined the performance of cooperative spectrum sensing using machine learning technique and wrote the paper; all authors had approved the final version.

References

[1] S. Haykin and P. Setoodeh, "Cognitive radio networks: the spectrum supply chain paradigm," *IEEE Trans. On Cognitive Communications and Networking*, vol. 1, no. 1, pp. 3-28, Oct. 2015.

[2] H. Huang, Z. Zhang, P. Cheng, and P. Qiu, "Opportunistic spectrum access in cognitive radio system employing cooperative spectrum sensing," in *Proc. of IEEE Vehicular Technology Conference (VTC'09- Spring 2009)*, April 2009, pp. 1–5.

[3] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Comm. Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, Mar. 2009.

[4] L. Jun, G. Zhang, and C. Yan, "An energy detection-based spectrum-sensing method for cognitive radio," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1-10, Feb. 2022.

[5] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201-220, Feb. 2005.

[6] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communications*, vol. 4, no. 1, pp. 40–62, Mar. 2011.

[7] D. Cabric, S. M. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. of 38rd Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2004, pp. 772-776.

[8] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental design tradeoffs in cognitive radio systems," in *Proc. of TAPAS'06*, Boston, MA, Aug. 2006, pp. 1-9.

[9] G. C. Sobabe, Y. Song, and B. Guo, "A cooperative spectrum sensing algorithm based on unsupervised learning," in *Proc. 10th Int., Cong. on Image and Signal Proces. BioMedical Engg. & Inform.*, Oct. 2017, pp. 1-6.

[10] Z. Han, R. Zheng, and H. V. Poor, "Repeated auctions with Bayesian nonparametric learning for spectrum access in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 10 (3), pp. 890–900, Jan. 2011.

[11] Nair, G. Resmi, and K. Narayanan, "Cooperative spectrum sensing in cognitive radio networks using machine learning techniques," *Applied Nanoscience*, pp. 1-11, Feb. 2022.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning J.*, vol. 20, pp. 273–297, Sep.1995.

[13] Atta-ur-Rahma, "Efficient decision based spectrum mobility scheme for cognitive radio based V2V communication system," *JCM*, vol. 13, no. 9, pp. 498-504, Sep. 2018.

[14] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1823–1834, Feb. 2010.

[15] O. P. Awe, D. A. Babatunde, S. Lambotharan, and B. AsSadhan, "Second-order kalman filtering channel estimation and machine learning methods for spectrum sensing in cognitive radio networks," *Wireless Networks*, vol. 27, pp. 3273-3286, May 2021.

[16] C. Prasad and P. Bojja, "A reliable, energy-aware and stable topology for bio-sensors in health-care applications," *Journal of Communications*, vol. 14, no. 5, pp. 390-395, May 2019.

[17] Q. Zhu, Z. Han, and T. Basar, "No-Regret learning in collaborative spectrum sensing with malicious nodes," in *Proc. 2010 IEEE International Conference on Communications*, Cape Town, South Africa, May 2010, pp. 1–6.

[18] A. Hassaan, "Ensemble classifier based spectrum sensing in cognitive radio networks," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1-16, Jan. 2019.

[19] H. Yasmin, E. Tarhuni, and A. Khaled, "Learning-Based spectrum sensing for cognitive radio systems," *Journal of Computer Networks and Communications*, vol. 2012, pp. 1-13, Dec. 2012.

[20] E. Waleed, U. Hasan Najam, A. Saleem, and K. Hyung Seok., "Fuzzy logic based spectrum sensing for cognitive radio networks," in *Proc. IEEE Fifth Int. Conf. on Next Gen. Mobile Applications Serv. and Tech.*, Sep. 2011, pp. 185-189.

[21] D. Suddhendu, P. Sharanya, N. S. Shivendra, and G. Mitali, "Performance analysis of prediction based spectrum sensing for cognitive radio networks," in *Proc. IEEE 2nd Int. Conf. on Intelligent Communication and Computational Techniques*, Sep. 2019, pp. 271-274.

[22] K. MadushanThilina, C. K. Won, S. Nazmus and H. Ekram, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2209-2221, Oct. 2013.

[23] C. Chhagan and R. Pandey, "Intelligent selection of threshold in covariance-based spectrum sensing for cognitive radio networks," *Wireless Networks*, vol. 24, no. 8, pp. 3267-3279, Nov. 2018.

[24] L. Zhihan, Z. Youjian, X. Haowen, C. Wenxiao, X. Shangqing, L. Yilin, and P. Dan, "Unsupervised clustering through gaussian mixture variational AutoEncoder with

non-reparameterized variational inference and std annealing," in *Proc. IEEE International Joint Conference on Neural Networks*, Jul. 2020, pp. 1-8.
[25] F. Chris and R. Adrian, "Model-Based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611-631, Jun. 2002.

Srinivas Samala received his M. Tech, a degree from the JNTU Hyderabad, India. He is currently pursuing a Ph.D. degree with the School of Electronics Engineering, KIIT Deemed to be University, Bhubaneswar, India. His research interests include cognitive radio and networks, spectral estimation, signal processing.

Subhashree Mishra has 9 years of experience in teaching. She has completed her Ph.D. in 2017. Her specialization includes mobile communication, cognitive radio, and networking. She has published many research papers in reputed journals and conferences. She has published three book chapters in Springer as her credit.

Dr. Sudhansu Sekhar Singh has received a Ph.D. in Engineering (Wireless Communication) from Jadavpur University, Kolkata, India, and an M.E. in Electronic System and Communication Engineering from R.E.C (presently NIT) Rourkela, India. He is working as a Professor at the School of Electronics Engineering, KIIT University, Bhubaneswar, India. He has more than 20 years of teaching experience and 10 years of other professional experience. More than 75(seventy-five) publications in international journals and reputed international conference proceedings are to his credit. Also, he has guided more than twenty-five PG thesis and Three doctoral theses. He has also examined several doctoral dissertations. His broad research area includes but is not certainly limited to wireless and mobile communication, multicarrier CDMA, MIMO-OFDM, Wireless Sensor Networks.