Intelligent Automation And Soft Computing, 2020
Vol. 26, no. 4, 817–830
DOI: 10.32604/iasc.2020.010116

Detection of the Spectrum Hole from N-number of Primary Users Using the Gencluster Algorithm

U. Venkateshkumar[1], S. Ramakrishnan[2]

[1]Department of ECE, Sri Krishna College of Technology,Coimbatore,641042,INDIA.
[2]Department of IT, Dr. Mahalingam College of Engineering and Technology,Pollachi,642003,INDIA.

ABSTRACT

A hybrid form of the genetic algorithm and the modified K-Means cluster algorithm forms as a Gencluster to detect a spectrum hole among n-number of primary users (PUs) is present in the cooperative spectrum sensing model. The fusion center (FC), applies the genetic algorithm to identify the best chromosome, which contains many PUs cluster centers and by applying the modified K-Means cluster algorithm identifies the cluster with the PU vacant spectrum showing high accuracy, and maximum probability of detection with minimum false alarm rates are achieved. The graphical representation of the performance metric of the system model shows 95% accuracy with 98.63% detection probability and 1.36% false alarm probability.

KEYWORDS: Clustering, cooperative spectrum sensing, fusion center, genetic algorithm, signal to noise ratio.

1 INTRODUCTION

IN the twenty first century, the innovation of mobile phones made a revolution in the telecommunication industry. In 2003, Andy Rubin, Rich Miner, Nick Sears, and Chris White started the Android Corporation in California. They described the potential of the android operating system (OS) in developing the smartphone. The Android OS made a big revolution in the smartphone usage with various Apps (applications). As per the Nokia India mobile broadband index (The Times of India, 2019) released in 2019 predicted, 432 million 4G users, which is a 137% increase when compared to 2017 as shown in Figure 1, and each data user's downloading is about 10Gb on an average per month. The TRAI (Telecom Regulation Authority of India) report (TRAI, 2018) shows that above 500 million mobile subscribers are in India, and the telecom service revenue contributed about 1.97% of the GDP (Gross Domestic Product) growth in India, for the financial year 2016-17. The Nokia India broadband index in Figure 2 shows that live video streaming is the choice of data opted by 80% of the consumers. YouTube, Hotstar, Jio TV, Skype, Google Duo, and Netflix are the various Apps that support video calling and live video streaming, which consumes have a large bandwidth, which in turn creates huge demand for the spectrum. In India, for the past 160 years the telegraph service is used, but in the modern world people are not using this service effectively, so the government of India has decided to stop using the telegraph service as or July 15, 2013. A report prepared by the Spectrum-Policy Task Force on improving the utility of available spectrums effectively in the United States was published by the Federal Communication Commission (FCC) November 2002 Marcus, et.al. (2002). The report shows that, the majority of the spectrum band is unutilized, so researchers focused on techniques to use the unused spectrum effectively. In 1999, Joseph Mitola III, discussed the “Pooled Radio Spectrum" (Mitola, 1999). Simon Haykin, (Haykin, 2005) coined the word spectrum hole or white space and he proposed the novel idea to utilize the available natural resource of the electromagnetic spectrum and challenges were faced in building this model. Various methods of the spectrum sensing are reviewed in (Haykin, 2005, Akyildiz, et.al. 2011), which includes matched filtering, energy detection, cyclostationary, cooperative spectrum sensing, hardware challenges and different standards employed in sensing are discussed briefly. In this paper, the following contributions are discussed.

CONTACT U. Venkateshkumar ✉ venky.2019@outlook.com

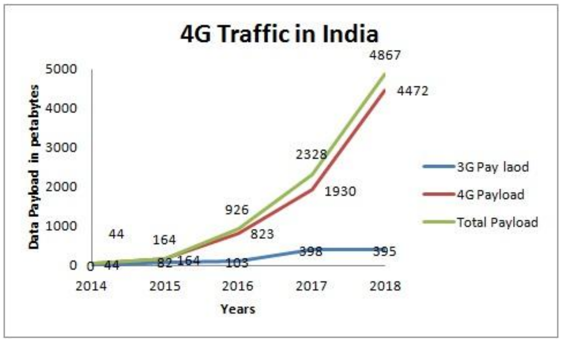


Figure 1. **4G Traffic in India.**

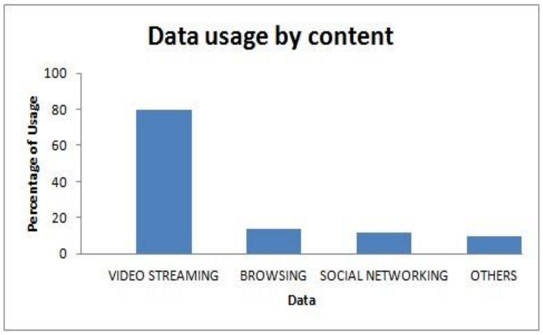


Figure 2. **The Data usage by content opted by consumers.**

A). A cooperative spectrum sensing (CSS) system model, along with the Fusion Centre (FC) detects the spectrum hole among the n-number of primary users (PU's) is framed.

B). The genetic algorithm is applied to select the best chromosome, which contains n-number of cluster heads.

C). An unsupervised machine learning (ML) algorithm namely, and the modified K-means (MK-Means) cluster algorithm is applied to detect the spectrum hole.

D). The Adaboost supervised classification algorithm is applied to the custom-made dataset of 600 training samples to detect the spectrum hole.

E). A test set of 120 PUs are tested using the genetic algorithm.

This paper is organized as follows, Section 2 provides an overview of the conventional CSS and the energy detection techniques used in the spectrum sensing and further is described regarding the proposed CSS system model using the k-means cluster algorithm. Section 3, represents, a comparative report of two ML methods namely the RF classification and the k-means clustering algorithm. Section 4 represents the simulation results with the performance metric curve and Section 5 concludes the paper.

2 THE SYSTEM MODEL

2.1 Conventional Framework of the Cooperative Spectrum Sensing model.

A conventional CSS model (Haykin, 2005) is shown in Figure 3. It consists of a FC and one or many cognitive radio (CR) users or secondary users (SU). The CR users sense the transmission of the PU, at every instant of time, and then passes the sensing information through the control channel to the FC. The FC receives the sensing information from all CR users through the control channel and determines the presence or absence of the PU's and conveys the back to all CR users. The sensing task performed by each CR user is called local sensing. Consider the received signal as y(t) and the transmitted signal as x(t), h(t) is the channel gain of the sensing channel and n(t) as the zero mean additive white gaussian noise. The PU signal detection can be formulated by the binary hypothesis as shown in equation 1.

$$y(t) = \begin{cases} n(t), & H_0 \\ h(t)\cdot x(t) + n(t), & H_1 \end{cases} \qquad (1)$$

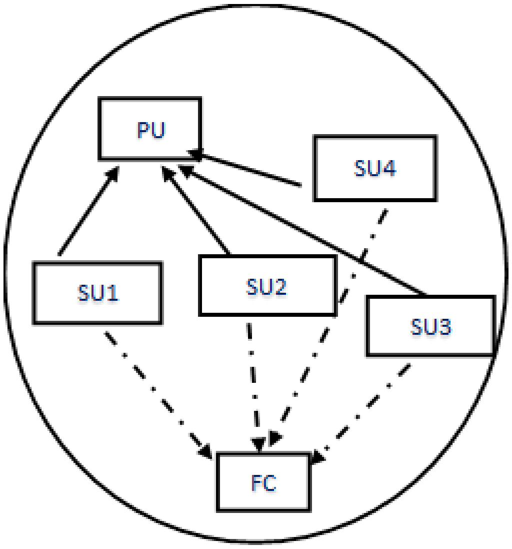


Figure 3. **The Conventional Cooperative sensing model.**

H_0 and H_1 indicates the absence and presence of the PU's respectively. The energy detection technique shown in Figure 4 decides the PU presence or absence, by comparing the PU signal level with the threshold value. If the PU signal level is less than the threshold, it shows that the PU is vacant and if it is greater than the threshold, then the PU is busy.

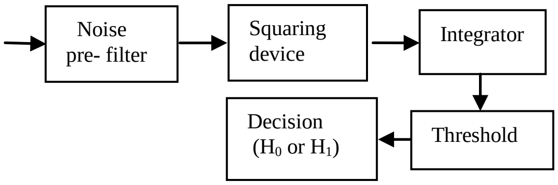


Figure 4. **The Conventional Energy Detection Technique.**

A fuzzy c-means clustering algorithm is shown in Chatterjee, et.al. (2014) where multiple SU's sense the PU simultaneously and pass this information to a FC using the control channel. FC and categorize the PU presence into four different clusters using fuzzy c-means clustering algorithm as shown in Figure 5. The dynamic threshold, along with the energy-efficient spectrum sensing and a few advanced methods in the CSS are discussed in Safatly, et.al. (2014), Hassan, et.al. (2012), Zhao, et.al. (2013), Hossain et.al. (2012), Noorshams, et.al. (2010), Salahdine, et.al. (2015), You, et.al. (2017).

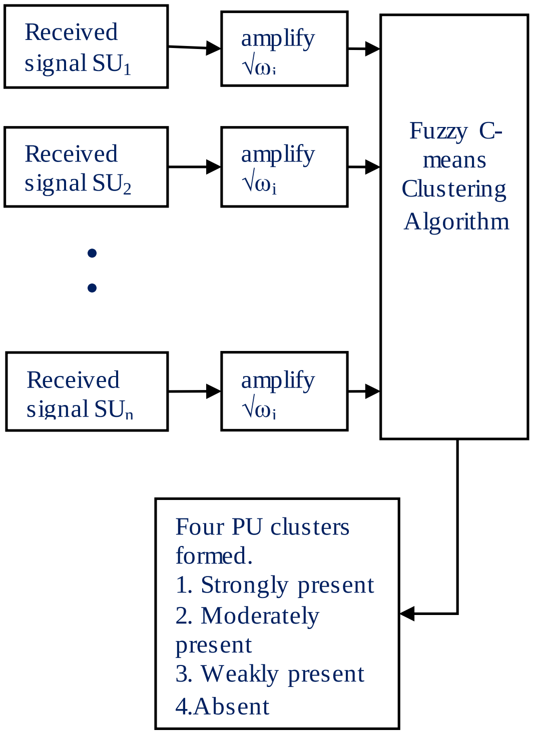


Figure 5. **The Spectrum Sensing using Fuzzy C-means Cluster Algorithm.**

Table 1. **The Clusters Formed using the K-means Algorithm for Different Test Samples.**

S.No	Number of PU's test samples	Number of cluster heads formed	Number of iteration	Members of cluster heads			
				T	F	T	F
1	10	2	4	6	4	-	-
2	20	2	4	11	9	-	-
		3	8	5	9	6	-
3	30	2	4	16	14	-	-
		3	8	8	14	8	-
4	40	2	4	19	21	-	-
		3	8	10	21	9	-
5	50	2	4	19	31	-	-
		3	8	10	31	9	-
6	60	2	4	28	32	-	-
		3	8	15	32	13	-
		4	9	15	31	12	2
7	70	2	4	29	41	-	-
		3	8	16	41	13	-
		4	9	16	40	12	2
8	80	2	4	33	47	-	-
		3	8	19	47	14	-
		4	9	19	46	13	2
9	90	2	4	37	53	-	-
		3	8	21	53	16	-
		4	9	21	52	15	2
10	100	2	4	40	60	-	-
		3	8	22	16	18	-
		4	9	22	59	17	2
11	110	2	4	43	67	-	-
		3	8	25	67	18	
		4	9	25	66	17	2
12	120	2	4	48	72	-	-
		3	8	28	72	20	-
		4	9	28	71	19	2

2.2 *Limitations of the K-means Cluster Algorithm*

The clustering method is used to identify the hidden information from the unknown data, and it is used to discover the relationship among the data. The clustering algorithm, divides the data into groups with similar instances as together framing a single cluster. The clusters are separated by dissimilar features or properties and the k-means algorithm requires a user to initialize the number of clusters to be formed. The algorithm randomly selects the initial input seeds for the cluster formation. But in real time, a user may not have enough knowledge on the number of clusters to be formed, because it varies with respect to different data sets and similarly randomly selecting the initial seeds will affect the cluster formation at the end. Another limitations are it does not store the high-quality clusters formed in a previous iteration and it fails to progress when the local minima forms poor quality clusters. The k-means clustering algorithm is applied to a custom based dataset of 10,20, …120 instants with six attributes which includes signal to noise ratio (SNR) values, threshold, distance, probability of false alarm (Pfa) and PU_vacant class. The PU_vacant class indicates the presence or absence of the spectrum hole in our system model. The FC sensing these n-number of PU's using the k-means clustering algorithm is performed and the Table 1 shows the different test samples with different numbers of cluster heads and corresponding cluster members forming the PU_vacant class with true (T) and false (F) readings. The table shows when the number of test samples are 60 and above the results are compared with three and four cluster heads. Under three cluster heads, 13 users are considered as the PU_vacant true class and the same 60 samples with four cluster heads Twelve users are considered as the true class and one user is added to the PU_vacant false class. This shows that a misdetection takes place by initializing the number of clusters in the k-means and the random selection of data from the data set for the cluster formation, which predicts a poor cluster formation.

2.3 The Proposed CSS System Model

Shown in Figures 6 is the proposed CSS model, showing the n-number of PU's sensing process by the FC. The sensed signal is passed through the band pass filter (BPF), where each PU with different frequency bands are identified. Further, each PU's received signal power is calculated and a pre-threshold value is added to form a n-number PU's dataset. To identify the PU presence or absence using the clustering algorithm is processed using this dataset. The genetic algorithm is applied to identify the initial population size and best chromosome selection for the cluster formation. Later using the initial population and best chromosome, the PU presence is identified using the MK-means clustering algorithm. This process is repeated for different populations of the PU's to identify its cluster head. The detailed algorithm process is discussed in Section 3. The proposed CSS system model is shown in Figure 7. Assume the n-number of PUs are sensed by the FC during the sensing time interval T_s. The channel link established in the network would be FC to PU_1, FC to PU_2, FC to PU_3, and FC to $PU_{N.}$ During the sensing time, sensing interval T_s is divided into the N nonoverlapping slots, each duration of $t_{s,}$ to sense the n-number of PU's. The signal received by the fusion centre from the n-number of PUs is given by equation 2.

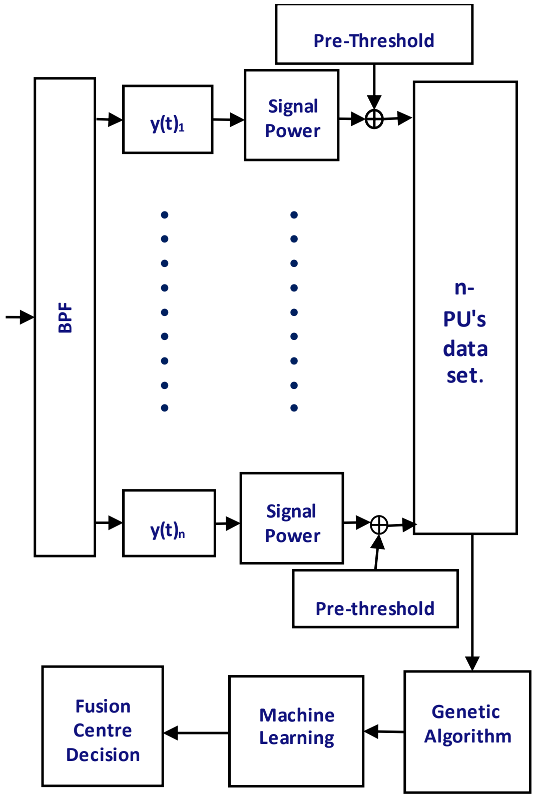


Figure 6. Proposed Cooperative Spectrum Sensing System Model.

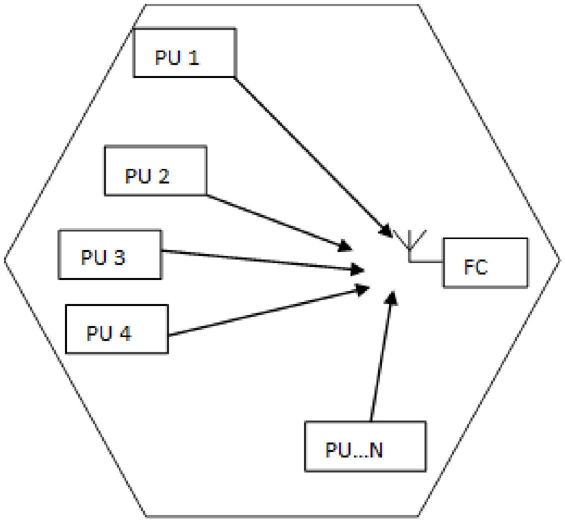


Figure 7. The Proposed CSS System Model.

$$y(n) = \sum_{n=1}^{N} h_{nc} x(n) + v(n) \quad \forall n = 1....N \qquad (2)$$

where v(n) is the noise component added with the original signal x(n) transmitted from each PU. It is assumed that all n-number of the PU's signals received are mutually independent i.e. of x(n) ≠ x(i) for n ≠ i.

2.3.1 The Proposed CSS System Model using the Genetic algorithm

The initial number of the population taken for the cluster formation and the defining number of clusters to be formed is automated using the genetic algorithm. The genetic algorithm flow diagram shown in Figure 8, to form the clusters from the best fit chromosome. The Initial number of chromosome formation process flow chart is shown in Figure 9. The modified k-means algorithm is applied to the normalized data set to form a set of solutions to identify the PUs that are vacant. The metric used to form the solution using the MK-Means is the Manhattan distance. The Manhattan distance between two PUs is given in equation 3.

$$d\left(PU_{i,}PU_{l}\right) = \frac{\sum_{j=1}^{t}\left|PU_{i,j}-PU_{l,j}\right|+\sum_{j=\mathrm{t}+1}^{m} d(PU_{i,j},PU_{l,j})}{|A|} \quad (3)$$

where |A| = the number of attributes.

This will form a set of solution, which is considered as the chromosome in the genetic algorithm. The number of chromosomes taken as the input to the clustering algorithm is a two-stage process. In the first stage, nine different k values are taken as the input to the MK-Means, $k = K_i^1$, where i=1,2...9, delivering a clustering solution with the K_i^1 gene or cluster centres. This process is repeated for five times to give 45 chromosomes. In the second stage, different values to be used as the input to the MK-Means delivers a clustering solution with $k = K_i^2$ as the input for the number of clusters. For each input, K_i^2 a random number 1 to $\sqrt{|D|}$ is taken as the input. For each of the nine different values of the K_i^2, it is repeated for 5 times and delivers 45 chromosomes. Combining the two stages, a total of 90 chromosomes are produced as the initial population P_I. There are five clusters denoted as $C_{k,1},\dots C_{k,5}$ arranged in descending order for fitness. A fitness value of T(K) is chosen from the i^th solution for k, where $k \in K$, and we take the average fitness of the fifth solution.

A value of k is chosen with the probability of $k = \frac{T(K)}{\sum_{k \in K} T(K)}$. The process is repeated ' I ' times to have initial population of P_I. The initial population of the chromosome is taken as the input to the genetic algorithm shown in Figure 10 and further processing steps are given below.

Crossover

Elitism

Mutation

Probabilistic Cloning

Chromosome selection

In the above-mentioned steps, the Crossover, Elitism and Mutation operations are most found in the genetic algorithm process. The probabilistic cloning and chromosome selection are the most important process applied in the genetic algorithm-based system model.

The Crossover Operation

In the crossover operation, all chromosomes formed in the initial step are sorted in descending order and a chromosome pair is taken for the crossover operation is formed. The best fit chromosome is selected as the first and the second chromosome is selected using the roulette wheel technique. The first chromosome is referred to as the reference chromosome and the second as the target chromosome.

The Chromosome re-arrangement operation takes place from all the clusters of the target chromosome. The cluster having the minimum distance with the first cluster of the reference chromosome is placed at the front in the rearranged version of the target chromosome. In the same way, the cluster of the target chromosome having minimum distance with the second cluster of the reference chromosome is then placed as the second term of the rearranged version. In this way the rearrangement of the chromosome takes place until all the target clusters are rearranged with respect to the reference chromosome.

The Elitism Operation

The best chromosome of the previous generation is stored in the elitism operation as the CR_h is compared with the chromosome of the current generation produced by the crossover operation and if the chromosome found in the current generation is found best when compared to previous generation then the elitism operation replaces the best chromosome CR_h.

The algorithm is given in Table 1.

The Mutation Operation

The mutation operation randomly changes the cluster of a chromosome to avoid unusual results. The mutation probability of the i^th chromosome in the current population is given in equation 4.

$$M_i = \begin{cases} \frac{f_{max}-fitness(CR_i)}{2(f_{max}-\bar{f})} fitness(CR_i) > \bar{f} \\ \frac{1}{2} fitness(CR_i) < \bar{f} \end{cases} \quad (4)$$

where $f_{\max} = \max_{CR_j \in P_I} fitness(CR_j)$ is the fitness of the best chromosome in the current population. $\bar{f} = \sum_{CR_{j\in PC_I}} fitness(CR_J)/[\![P_I]\!]$ is the average fitness of the chromosome in the current population. A chromosome selected for the mutation operation randomly changes the attribute of each cluster of the chromosome.

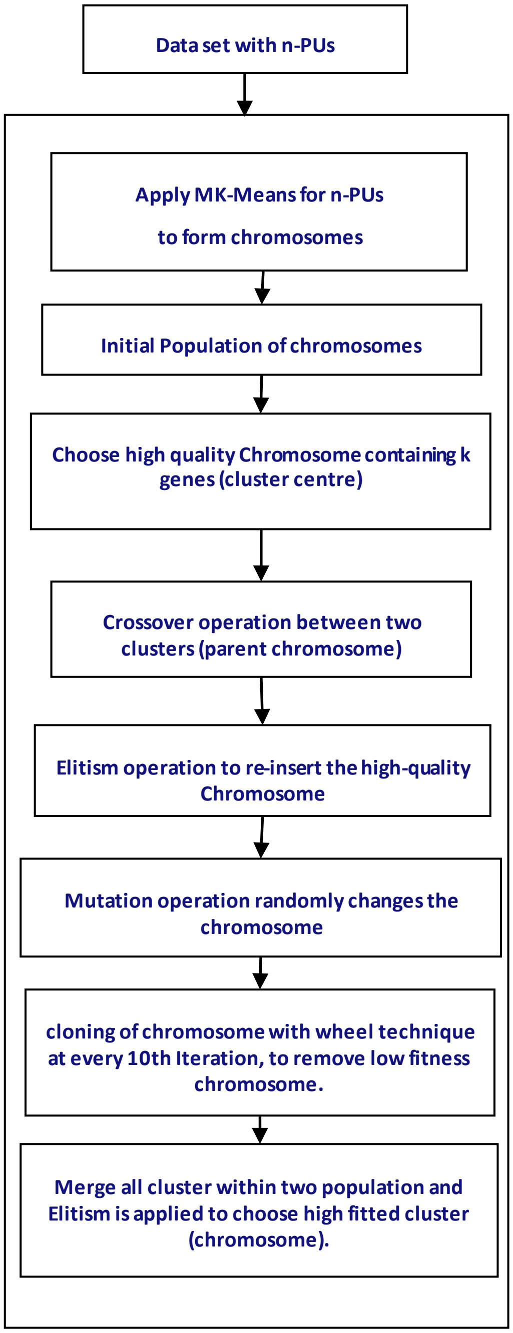

Data set with n-PUs
Apply MK-Means for n-PUs to form chromosomes
Initial Population of chromosomes
Choose high quality Chromosome containing k genes (cluster centre)
Crossover operation between two clusters (parent chromosome)
Elitism operation to re-insert the high-quality Chromosome
Mutation operation randomly changes the chromosome
cloning of chromosome with wheel technique at every 10th Iteration, to remove low fitness chromosome.
Merge all cluster within two population and Elitism is applied to choose high fitted cluster (chromosome).

Figure 8. The Genetic algorithm flow diagram

Table 1. Genetic Algorithm Forming Clusters

Input D: The data set which contains N number of PU's.
Initialise : Normalise the dataset.
Output: Formation of cluster C.
Step 1: Apply MK Means between two PUs to identify its clusters. cluster center of group of PUs is determined by average of all numerical attributes and mode of categorical attributes belonging to cluster.
Initial population of PUs chromosome is a two-stage process.
Stage I: Apply common PUs as input to K clusters and MK-Means delivers K_i^1 cluster. Each cluster forms a chromosome with K_i^1 genes.
Stage II: Apply different PUs as input to K clusters and it forms K_i^2 cluster. Apply different values of k random numbers between k =1 to √|D|.
Initial population $P_I = \sum_{k=1}^{9}\sum_{i=1}^{5} C_{k,i}$
Step 2: Choose high quality chromosomes (CR_h) by fitness operation.
Step 3: Arrange the high-quality chromosomes in order (L).
Step 4: Applying Crossover operation between parent chromosome, using highest value L and applying Roulette wheel technique for remaining L-1 value chromosome.
Step 5: Perform rearrange operation, because Chromosomes are list of cluster centers, and centers are our genes.
Step 6: Applying Elitism operation to re-insert the highest fitness best chromosome.
Step 7: Apply Mutation operation. Randomly changes the gene of a chromosome and randomly change the attribute of each gene of the selected chromosome. Finally, duplicate gene is removed.
Step 8: Cloning is performed after every 10 iteration and replace the high fitness chromosome.
Step 9: Chromosome selection merge all chromosomes between two population and delivering new population P_s.

The Cloning operation

Cloning is performed at every ten iterations. We generate a cloning population P_{CI} from the population P_I. The probability of a chromosome $CR_i \in P_I$ to be inserted into the cloning population $P_{CI} = \frac{fitness(CR_I)}{\sum_{j=1}^{|P_I|} fitness(CR_j)}$. The mutation is performed in all chromosomes in the cloned population generated by the P_{CI} in order to maintain dissimilarity among chromosome. The elitism is performed after every cloning operation, and gives the best chromosome found and is required to reach levels of high clustering quality.

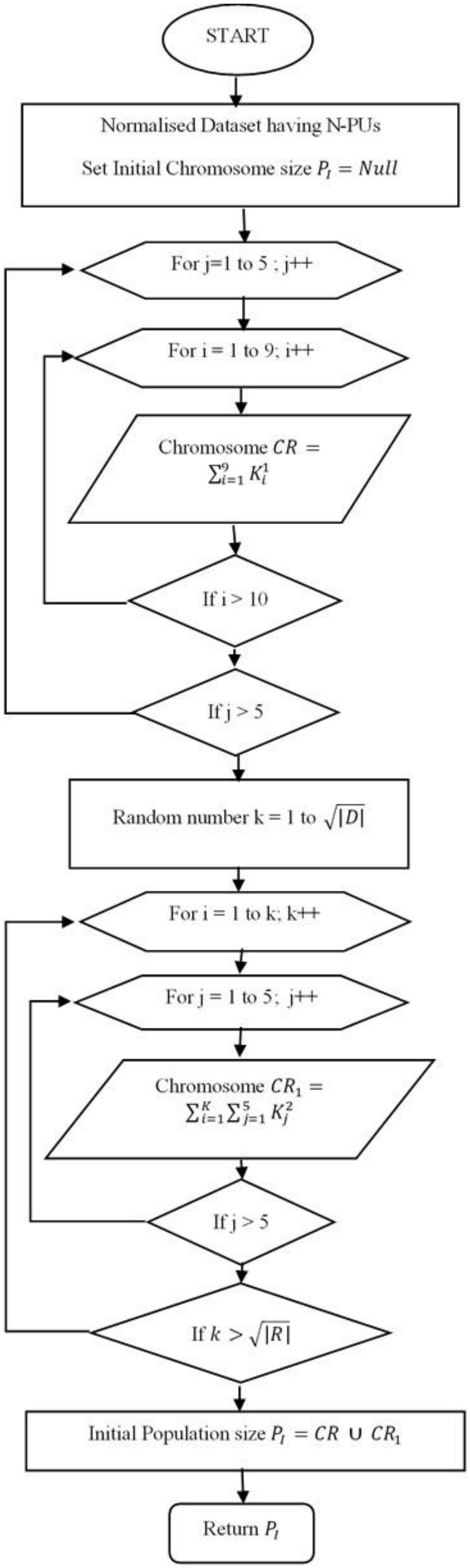
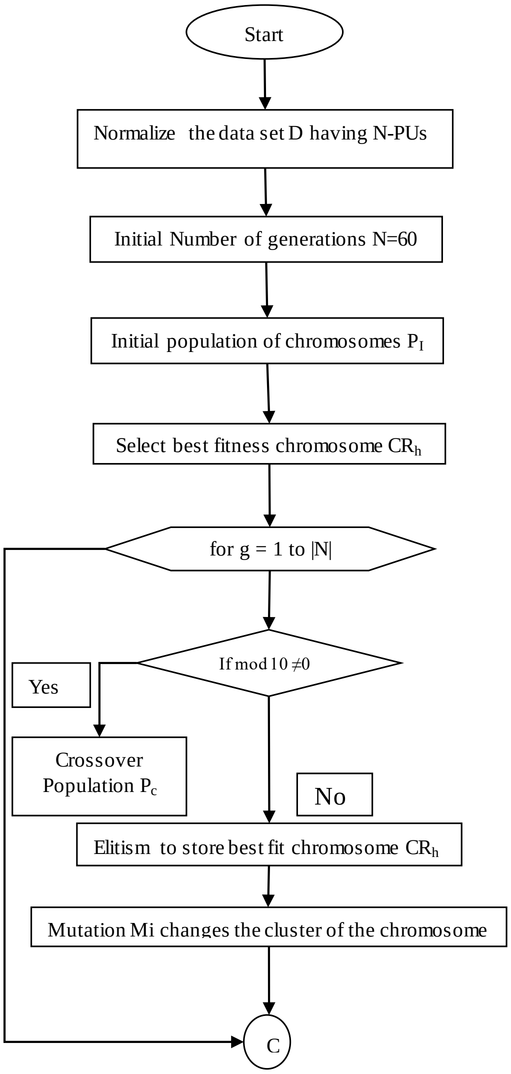

START
Normalised Dataset having N-PUs
Set Initial Chromosome size $P_I = Null$
For j=1 to 5 ; j++
For i = 1 to 9; i++
Chromosome $CR = \sum_{i=1}^{9} K_i^1$
If i > 10
If j > 5
Random number k = 1 to $\sqrt{|D|}$
For i = 1 to k; k++
For j = 1 to 5; j++
Chromosome $CR_1 = \sum_{i=1}^{K} \sum_{j=1}^{5} K_j^2$
If j > 5
If $k > \sqrt{|R|}$
Initial Population size $P_I = CR \cup CR_1$
Return P_I

Figure 9. The Initial Chromosome Formation Flow Chart


Start
Normalize the data set D having N-PUs
Initial Number of generations N=60
Initial population of chromosomes P_I
Select best fitness chromosome CR_h
for g = 1 to |N|
If mod 10 ≠0
Yes
Crossover Population P_c
No
Elitism to store best fit chromosome CR_h
Mutation Mi changes the cluster of the chromosome
C

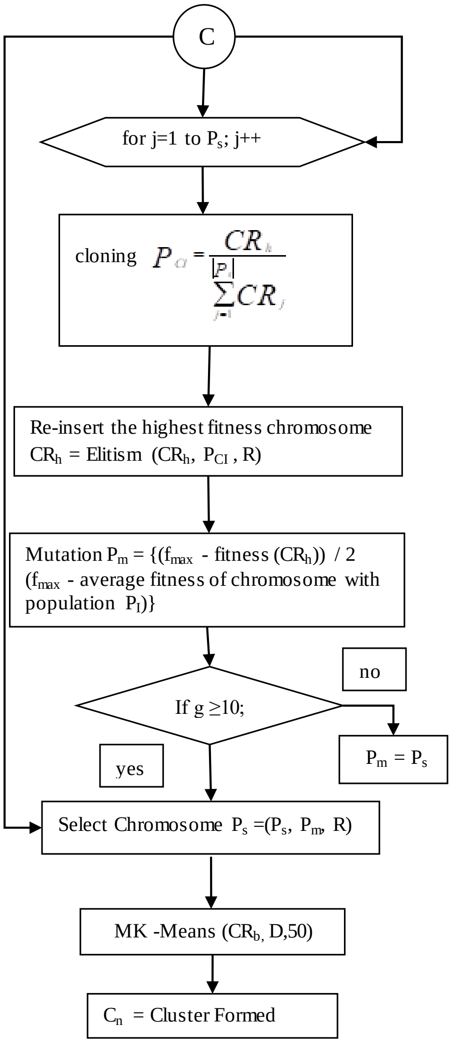

C
for j=1 to P_s; j++
cloning $P_{CI} = \frac{CR_h}{\sum_{j=1}^{|P|} CR_j}$
Re-insert the highest fitness chromosome
CR_h = Elitism (CR_h, P_{CI} , R)
Mutation P_m = {(f_{max} - fitness (CR_h)) / 2
(f_{max} - average fitness of chromosome with population P_I)}
no
If g ≥10;
P_m = P_s
yes
Select Chromosome P_s =(P_s, P_m, R)
MK -Means (CR_b, D,50)
C_n = Cluster Formed

Figure 10. **The Flow chart of the Cluster Formation using Genetic Algorithm.**

The Chromosome Selection

After cloning is applied in the tenth generation, the chromosome selection is applied between the two populations. The recent population and the resulting generation from all operations is performed. The elitism is applied to choose the best chromosome from the merged population.

The Improved System

The performance of the system is improved in the noise dominant region by selecting only the PU that has a negative SNR value in the initial population formation to form the cluster. This system identifies the spectrum hole in the negative SNR region.

3 MACHINE LEARNING ALGORITHMS

3.1 The Classification using Supervised Learning

THE ML learns from the previous output and makes predictions on the data. The ML doesn't follow the static instructions given by the programming. The ML algorithms are capable of learning from the training data and predicts the future behaviour of the system model. The machines take unbiased decisions, unlike human, so a computer learns from an unstructured dataset available to make predictions on the data, based on similarity, specific features reveal the category of future data to be tested. Two types of ML methods namely are,

Supervised ML

Unsupervised ML

The supervised ML method uses the Adaboost algorithm and the unsupervised ML method uses the MK-Means cluster algorithm to detect if the PU is vacant or not. Supervised ML is applied in a known system environment, where the trained samples are applied to predict the future values of the system as shown in Figure 11. In general, the knowledge about the CR system environment is unpredictable, so we use the unsupervised ML method to predict the future output of the system. The sensing parameters are collected as input attributes from an unknown RF environment. The input instances, with five different signal attributes are applied to a genetic algorithm, where a highest fitness chromosome is identified, which contains n-number of clusters. The MK-Means cluster algorithm is applied to identify the cluster with the PUs vacant and busy. The flow diagram of the unsupervised ML methods is shown in Figure 12.

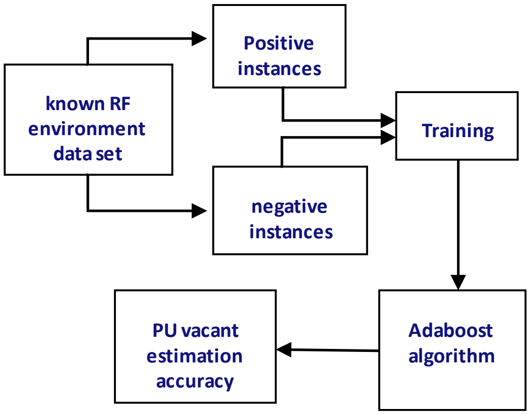


Figure 11. **The Supervised ML Flow Diagram**

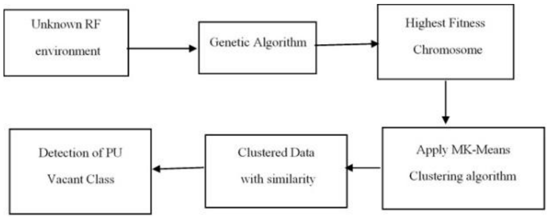

Unknown RF environment
Genetic Algorithm
Highest Fitness Chromosome
Apply MK-Means Clustering algorithm
Clustered Data with similarity
Detection of PU Vacant Class

Figure 12. The Unsupervised ML Flow Diagram.

3.2 The Adaboost Classification

The adaboost classification algorithm, classifies the input instances, into four different categories as follows

a. True Positive (TP)
b. False Positive (FP)
c. False Negative (FN)
d. True negative (TN)

The above-mentioned category forms a confusion matrix in RF algorithm and Table 2 shows a comparison between the confusion matrix formed. For the custom-made data set with 600 PU's training samples and sixty PUs test samples. Table 3 and Table 4 shows the summary result of the Adaboost algorithm using 600 PUs training samples with five different signal attributes and sixty PUs test samples. From the confusion matrix, the Adaboost classification algorithm classifies 600 training samples as the PU vacant and PU busy as true and false instances and is shown in Table 5. Performance of Table 6 indicates 2.8% of the training samples were misdetected as vacant and 9.5% mis-detected as busy. To predict the performance of the Adaboost algorithm, the test samples ranging from 10,20,30, to120 PUs were tested and the confusion matrix formed is given in Table 7. Table 8 shows the percentage of the misdetection, false alarm and detection accuracy with respect to the different PUs test samples. The predicted result table shows the false alarm percentage drops when the number of PU test samples increases and shows 98.36% accuracy for the 120 PU test samples with a false alarm just as 1.36% and the mis-detection increases with the increase in the PUs and shows a 89.3% accuracy for the 120 PU test samples with a 10.6% of mis-detection. The following performance metric graphical representation are discussed in the next simulation chapter.

Table 2: The Comparison Table between Training and Test Samples.

Algorithm: Adaboost algorithm Confusion Matrix					
600 Training Samples			Test Samples		
	Actual Instances Positive	Actual Instances Negative		Actual Instances Positive	Actual Instances Negative
Predicted output Positive	357 (TP)	10 (FP)	Predicted output Positive	26 (TP)	1 (FP)
Predicted output Negative	22 (FN)	211 (TN)	Predicted output Negative	1 (FP)	32 (FN)

Table 3. Summary of the Adaboost Classification using the Training Set of 600 Instances.

Correctly classified instances	568(94.6%)
Incorrectly classified instances	32 (5.3%)
Mean absolute error	0.0767
Root mean squared error	0.1979
Total number of instances	600
Relative absolute error	16.1422%
Root relative squared error	40.4357%

Table 4. Summary of Adaboost Classification using 60 PU's Test Sample Instances.

Correctly classified instances	58 (96.67%)
Incorrectly classified instances	2 (3.33%)
Mean absolute error	0.0617
Root mean squared error	0.1617
Total number of instances	60
Relative absolute error	12.7435%
Root relative squared error	30.9247%

Table 5. The Adaboost Classification Formed from the 600 Training Sample Instances.

S.NO	Classified Instances	Predicted output	600 Training samples classified	Number of PU mis-detected
1	True	PU vacant	357	10
2	False	PU busy	211	22

Table 6. The Adaboost Algorithm Performance Table using 600 Training Samples.

S.No	Classified Instances	Predicted output	Percentage of misdetection	Percentage of accuracy
1	True	PU vacant	2.8%	97.27%
2	False	PU busy	9.5%	90.5%

Table 7. **The Confusion Matrix Formed using the Adaboost Algorithm for the Different Test Samples.**

S.No	Number of PU's	Correctly classified instances	Incorrectly classified instances	TP	FP	FN	TN
1	10	10	0	6	0	0	4
2	20	20	0	11	0	0	9
3	30	20	0	16	0	0	14
4	40	40	0	19	0	0	21
5	50	50	0	19	0	0	31
6	60	58	2	26	1	1	32
7	70	68	2	27	1	1	41
8	80	78	2	31	1	1	47
9	90	87	3	34	2	1	53
10	100	96	4	36	3	1	60
11	110	105	5	38	4	1	67
12	120	114	6	42	5	1	72

Table 8. **The Predicted Misdetection, False alarm and Detection Accuracy for the Different Test Samples.**

S.No	Number of PU's Test Sample	Percentage of Misdetection	Percentage of False Alarm	Percentage of Detection	
		PU's vacant	PU's busy	PU's vacant	PU's busy
1	10	nil	nil	100%	100%
2	20	nil	nil	100%	100%
3	30	nil	nil	100%	100%
4	40	nil	nil	100%	100%
5	50	nil	nil	100%	100%
6	60	3.8%	3.03%	96.2%	96.9%
7	70	3.57	2.38%	96.4%	97.6%
8	80	3.1%	2.08%	96.8%	97.91 %
9	90	5.5%	1.8%	94.4%	98.14 %
10	100	7.6%	1.63%	92.3%	98.36 %
11	110	9.5%	1.47%	90.4%	98.52 %
12	120	10.6%	1.36%	89.3%	98.63 %

- Receiver operating characteristic (ROC) curve.
- Precision curve
- F-measure

3.3 The Clustering using the Gencluster++ Algorithm

The clustering algorithm divides data into groups with similar instances as together framing a single cluster. Clusters are separated by dissimilar features or properties. The input parameters are named as attributes in the ML and this includes the signal to noise ratio (SNR) values, the threshold, distance, the probability of false alarm (Pfa) and the PU_vacant class. The PU_vacant class indicates the presence or absence of the spectrum hole in our system model. Based on these five attributes, a data set is created with different PUs and a FC sensing these n-number of PUs using the Gencluster++ algorithm given in Table 9. The clusters formed using the Gencluster++ algorithm is compared with Table 1 in section 2 showing that when the number of test samples are 60 and above, the results are compared with three and four cluster heads using the k-means algorithm in Table 1. Under three cluster heads, 13 users considered as the PU_vacant class, as true and the same 60 samples with the four cluster heads are formed. The 12 users are considered as the true class and one user is added to the PU_vacant false class. This shows the misdetection take place when more number of PU's are considered, so in the k-means clustering algorithm, the initialization of the number of clusters to be formed leads to the misdetection and in the Gencluster++ algorithm, there is no need to initialize the number of cluster heads to be formed. Table 10 shows the performance of k-means cluster algorithm with the different test samples with the PU’s greater than 50. From Table 9, if the number of PU's is greater than 50, there is a no possibility of misdetection using the Gencluster++ algorithm. Table 10 shows, using the k-means that the percentage of the misdetection is 5% for the 120 PUs considered for the detection of the spectrum hole. The accuracy of detecting the PU_vacant class is about 95% for the PUs with 120 instances.

4 SIMULATION RESULTS

IN this section, we compare the graphical performance metrics, Boutaba, et.al. (2018), Abbas, et.al. (2015), obtained using the proposed Gencluster++ algorithm and the Adaboost classification algorithm.

Table 9. The Clusters Formed using the Gencluster++ Algorithm for Different Test Samples.

S. No	Number of PUs	Number of iterations	Number of clusters formed	F	F	T	T	F	T
1	10	2	3	1	3	6	-	-	-
2	20	2	4	1	5	11	-	3	-
3	30	2	4	1	8	16	-	5	-
4	40	2	4	1	12	19	-	8	-
5	50	2	4	1	20	19	-	10	-
6	60	2	4	2	21	27	-	10	-
7	70	2	4	2	26	28	-	14	-
8	80	2	4	2	26	32	-	20	-
9	90	2	4	2	30	36	-	22	-
10	100	2	4	2	34	39	-	25	-
11	110	2	4	2	38	42	-	28	-
12	120	2	4	2	41	47	-	30	-

Table 10. The K-means Algorithm Cluster Cells Formed from the Different Sets of the PU Test Samples.

No of PU's test sample	Number of clusters formed	Percentage of misdetection	Percentage of false alarm	Percentage of detection	
		PU's vacant	PU's busy	PUs vacant	PUs busy
60	3	7.6	nil	92.3	100
70	3	7.6	nil	92.3	100
80	3	7.14	nil	92.8	100
90	3	6.25	nil	93.75	100
100	3	5.55	nil	94.44	100
110	3	5.55	nil	94.44	100
120	3	5	nil	95	100

4.1 The Adaboost Classification Method Results

The performance metrics of the supervised ML using the Adaboost algorithm is discussed Table 11 and 12. From the performance metric table, for the different PU's test sample accuracy, the value is calculated and given in Table 13. The TPR mentioned in Table 11 gives you the ROC curve. First, a custom-made training samples of 600 instances is applied. The ROC curve plotted from the confusion matrix between the false positive rate and the true positive instances of the 600 training samples is shown in Figure 13. To predict the performance of the Adaboost algorithm, the different test samples ranging from 10, 20, to 500 PU's are applied and the ROC curve for all the test samples are shown in Figures no 14,15 and 16. The F-measure curve for the TPR is given in Figure 17.

Table 11. The Performance Metrics using the RF Algorithm.

S.No	Performance metrics	Description
1.	Accuracy	True positive predicted among total number of predictions
2	True positive rate (TPR) or Recall	True positive that are correctly predicted as positive.
3	False positive rate (FPR)	True negative predicted as positive.
4	Precision	Positive predictions that are correctly predicted.
5	F-measure	Trade-off between precision and recall

Table 12. The Performance Metrics Formula for System Validation

S.No	Performance metrics	Formula
1	Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
2	True positive rate (TPR) or Recall	$TPR = \frac{TP}{TP+FN}$
3	False positive rate (FPR)	$FPR = \frac{FP}{FP+FN}$
4	Precision	$Precision = \frac{TP}{TP+FP}$
5	F-measure	$F - measure = 2\frac{Recall \cdot Precision}{Recall + Precision}$

Table 13. The Calculation of the Accuracy for the Different Test Samples using the Adaboost Algorithm.

S.NO	Number of PU's test samples.	Percentage of Accuracy
1	10	100.00
2	20	100.00
3	30	100.00
4	40	100.00
5	50	100.00
6	60	97.5
7	70	97.5
8	80	97.5
9	90	96.67
10	100	96.00
11	110	95.45
12	120	95.00

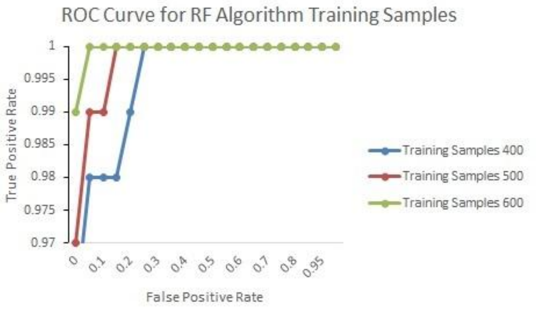


Figure 13. The ROC curve for the 600 Training Samples.

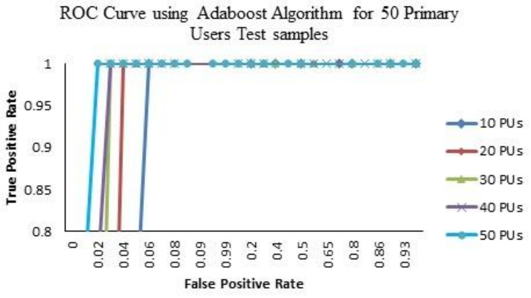


Figure 14. The ROC Curve for the PUs Ranging from 10 to 50 Users.

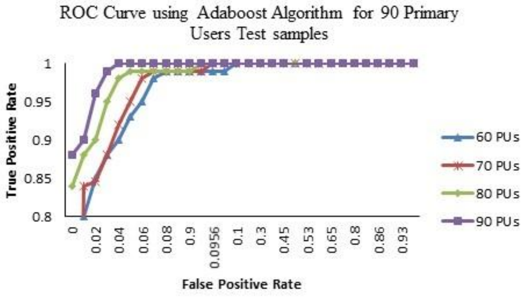


Figure 15. The ROC curve for the PUs Ranging from 60 to 90 Users.

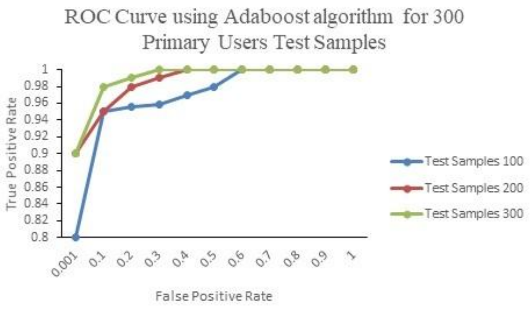


Figure 16. The ROC curve for the PUs Ranging from 100 to 300 Users.

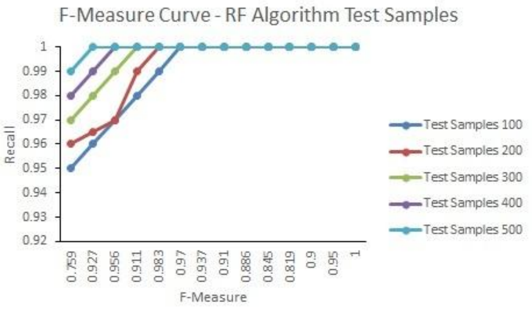


Figure 17. The F-measure Curve for the 500 PU's Test Sample.

4.2 The Comparison between the k-means and the Gencluster Simulation Results

Our proposed Gencluster algorithm is applied using the Weka tool as mentioned in Islam, et.al. (2018) and is compared with the k-means cluster algorithm to identify the spectrum hole detection. For simulation purposes, the FC sensing different sets of the PU test samples are considered given in Table 9. Ten PU test samples forming clusters is given in Figure 18 and hundred PUs forming four clusters with its cluster members is shown in Figure 19. For the 500 test samples, 6 clusters are formed as shown in Figure 20. As mentioned in Table 9, for the 120 test samples, 5 clusters are formed as shown in Figure 20. When the k-means cluster algorithm is applied to the 120 PU's, 3 clusters are initialized and one PU is mis-detected and identified as the 9th PU shown in Figure 21. The same test set with the 60 PUs is applied, forming three cluster heads with one user mis-detected is shown in Figure 22. Using the Gencluster algorithm, the cluster is formed based on the initial population so there is no need to initialize the number of clusters to be formed at the initial stage. This shows better performance than the k-means clustering algorithm. A comparison table between the k-means and the Gencluster algorithms is applied in our system model shown in Table 14.

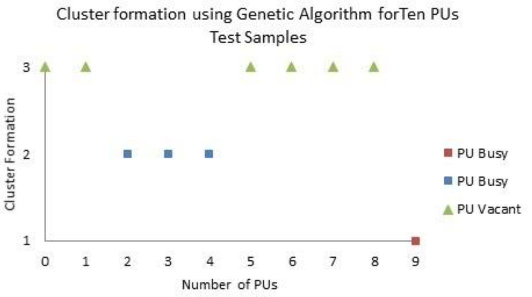


Figure 18. Ten PU Test Samples Forming Clusters.

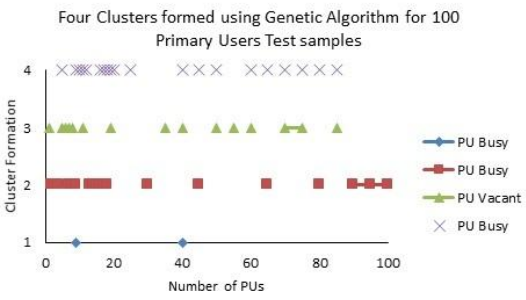


Figure 19. The PU's with 100 Test Samples Forming Clusters

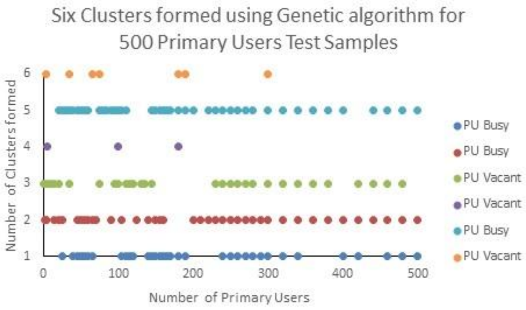


Figure 20. The PU's with 500 Test Samples Forming 6 Clusters.

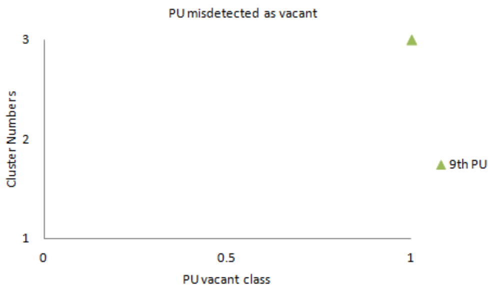


Figure 21. The 9th PU is Mis-detected as Vacant from the 120 Test Samples when 3 Clusters are Formed.

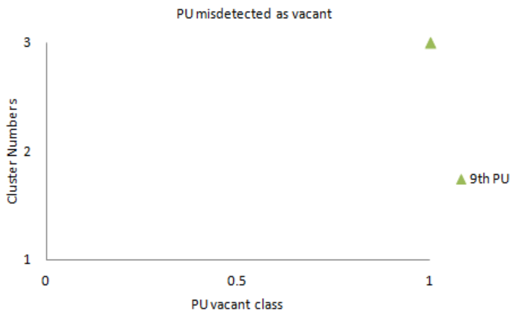


Figure.22. The 9th PU is Mis-detected as Vacant from the 60 Test Samples when 3 Clusters are Formed.

Table 14. The Comparison of Proposed Gencluster Algorithm.

S.NO	Performance Metrics	K-Means cluster algorithm	Gencluster algorithm
1	Accuracy	98.33% for 3 cluster formation with 120 PUs test samples. SNR range (10dB to -12dB)	95% for 120 PUs test samples.
2	Probability of detection (P_d)	95% for 120 PUs test samples with 3 cluster heads.	100% for 120 PUs test samples. Cluster formation is automated.
3	Probability of false alarm (P_{fa})	nil	nil
4	Multiple PU sensing	yes	yes

5 CONCLUSION

THE proposed CSS system model using the Genetic algorithm forming clusters highlight the accuracy of the detecting spectrum hole without any misdetection for the 120 PU test samples. The Adaboost algorithm is used to classify the PU presence in a CSS system. The Classification with the 600 training samples and the 120 PU test samples shows a 95% accuracy. The applied Gencluster algorithm without any prior knowledge of the system selects the initial population for the cluster formation and detects the spectrum hole from the best selected chromosomes from 120 PU test samples with 100% accuracy without any misdetection. The system can be further improved by selecting the PU in the negative SNR region to detect the spectrum hole in the noise dominant region. The work can be extended with scheduling the identified spectrum hole to different SUs in a cognitive radio system.

6 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

7 REFERENCES

Abbas N, Nasser Y, El Ahmad K, 'Recent advances on artificial intelligence and learning techniques in cognitive radio networks', EURASIP Journal on Wireless Communications and Networking, 2015, (1),pp.174.

Akyildiz IF, Lo BF, Balakrishnan R. 'Cooperative spectrum sensing in cognitive radio networks: A survey', Physical communication, 2011,1,4(1), pp.40-62.

Boutaba R, Salahuddin MA, Limam N, Ayoubi S, Shahriar N, Estrada-Solano F, Caicedo OM, 'A comprehensive survey on machine learning for networking: evolution, applications and research opportunities', Journal of Internet Services and Applications, 2018, 9(1), pp.1-16.

Chatterjee S, Banerjee A, Acharya T, Maity SP, 'Fuzzy c-means clustering in energy detection for cooperative spectrum sensing in cognitive radio system', International Workshop on Multiple Access Communications, 2014,pp. 84-95.

Hassan Y, El-Tarhuni M, Assaleh K. 'Learning-based spectrum sensing for cognitive radio systems', Journal of Computer Networks and Communications, 2012.

Haykin S. 'Cognitive radio: brain-empowered wireless communications', IEEE journal on selected areas in communications, 2005 Feb 7, 23(2), pp.201-220.

Hossain MA, Hossain MS, Abdullah MI, 'Performance analysis of cooperative spectrum sensing in cognitive radio', International Journal of Innovation and Applied Studies, 2012,1(2), pp.236-45.

Indians now use 12 times more mobile data than in 2015, (2019, February 28) The Times of India, p.1.

Islam, Md Zahidul, Vladimir Estivill-Castro, Md Anisur Rahman, and Terry Bossomaier, "Combining k-means and a genetic algorithm through a novel arrangement of genetic operators for high quality clustering." *Expert Systems with Applications*,91, (2018), pp: 402-417.

Marcus M, Burtle J, Franca B, Lahjouji A, McNeil N. "Report of the unlicensed devices and experimental licenses working group". (Federal Communications Commission Spectrum Policy Task Force. Nov.2002), pp.1-24.

Mitola, Joseph. "Cognitive radio for flexible mobile multimedia communications." IEEE International Workshop on In *Mobile Multimedia Communications, (MoMuC'99) 1999,* pp.3-10.

Noorshams N, Malboubi M, Bahai A, 'Centralized and decentralized cooperative spectrum sensing in cognitive radio networks: A novel approach', IEEE Eleventh International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2010, pp.1-5.

Safatly L, Bkassiny M, Al-Husseini M, El-Hajj A., 'Cognitive radio transceivers: RF, spectrum sensing, and learning algorithms review', International Journal of Antennas and Propagation, 2014, pp.1-21.

Salahdine F, El Ghazi H, Kaabouch N, Fihri WF, 'Matched filter detection with dynamic threshold for cognitive radio networks', IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM), 2015, pp. 1-6.

TRAI, 'Telecommunication in select countries polices-statistics', (TRAI, November, 2018), pp.5-23.

You, Xiaojian, Xiaohai He, and Xuemei Han. "A Novel Solution to the Cognitive Radio Decision Engine Based on Improved Multi-Objective Artificial Bee Colony Algorithm and Fuzzy Reasoning." Intelligent Automation & Soft Computing, Vol.23, no.4, 2017, pp.643-651.

Zhao N, Yu FR, Sun H, Nallanathan A, 'Energy-efficient cooperative spectrum sensing schemes for cognitive radio networks', EURASIP Journal on Wireless Communications and Networking, 2013,(1),pp.120.

8 NOTES ON CONTRIBUTORS

U. Venkateshkumar received B.E. (ECE) in 2004, and M.E. (CS) in 2007 from Anna University Chennai, India. He is an assistant professor in the Department of ECE, Sri Krishna College of Technology, Coimbatore. He has 10 years of teaching experience. His area of research includes cognitive radio

S. Ramakrishnan received B.E. (ECE) in 1998, M.E. (CS) in 2000 and Ph.D. degree from Anna University, Chennai in 2007. He is a professor and the Head of IT Department, Dr. Mahalingam College of Engineering and Technology, Pollachi. His areas of research include digital image processing, information security and soft computing.