CHAPTER 5

HYBRID ANFIS SPECTRUM SENSING METHOD FOR COGNITIVE RADIO NETWORKS WITH MALICIOUS USER DETECTION

5.1. Introduction

The previous chapter clearly explained the spectrum sensing based on Fuzzy Neural Network (FNN). It must be observed that all rules are not guaranteed to be coherent. Hence, it is possible to have a mismatch between the rules. The aforementioned factor is considered as an important drawback of the FNN. If the mismatch between the rules occurred, then the primary user state is also misclassified. The spectrum opportunities may be overlooked resulting in inefficient spectrum utilization. As a result, the collision occurs if the secondary user starts using the identified spectrum band based on the primary user state for communication. In this chapter, this problem is clearly overcome by using the clustering method. The proposed method in this chapter consists of three phases, they are spectrum segmentation, clustering cognitive radio nodes and spectrum sensing.

The first phase is spectrum segmentation as discussed in fourth chapter; spectrum segmentation process is used for identifying the boundaries of the sub-band and then it is given as an input to the classifier. The second phase in the proposed approach is clustering. The Cognitive Radio (CR) nodes or secondary users are clustered or grouped based on the location parameter. The location parameter is identified for each and every node in the network, and the neighboring nodes are grouped together. Initial spectrum sensing is started once the group formation is finished. In order to select the cluster head, initial spectrum sensing is used. The node with most reliable sensing result will be considered as Cluster Head (CH) in the cluster. Hybrid ANFIS
algorithm is used for spectrum sensing. The clustering, clustering approaches, and hybrid ANFIS are analyzed thoroughly in this chapter.

5.2. Basic concept of clustering

Clustering is an unsupervised grouping process that identifies an inherent structure within a dataset by classifying data into different groups called clusters, and detailed processed of clustering is explained by Yeung et al., (2005). Clustering is done with no prior information about the intrinsic grouping, and all the grouping is data driven. Thus it is unsupervised process. This means no training data is used in the clustering process.

The main goal of clustering is to determine the intrinsic grouping in a dataset. The intrinsic grouping within a dataset is done by defining some criteria to be used for clustering. As a result, objects in the same cluster are nearly same as each other, than objects in different clusters according to some defined criteria. A typical grouping of data points with clustering in x-y coordinate system is shown in Figure 5.1. The points are clustered into three different groups based on the location of the points in the x-y coordinate system. It could be seen from the figure that points which are very close to each other belong to the same cluster. In this example, the Euclidean distance between the points is taken as measure of the criterion in clustering.

The criteria to be chosen are expected to provide the best grouping within the dataset. This depends on the application that clustering is to be utilized. Accordingly, the criterion that best fits a data set for a good clustering is determined by the user according to the application requirement. Clustering is used in different applications including the process of finding representative for homogenous groups (data reduction), finding useful and suitable groupings (data classes) or finding unusual data objects (outlier detection). In this chapter, the technique of clustering is employed to find suitable groupings of nodes in the proposed node selection method.
Since clustering is the grouping of similar instances/objects, some sort of measure that can determine whether two objects are similar or dissimilar is required. There are two main types of measures or criteria are used to estimate this relation: distance measures and similarity measures.

**Distance measure**

Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances \( x_i \) and \( x_j \) as: \( d(x_i, x_j) \). A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors. Many distance measures have been proposed in literature for data clustering. Most often, these measures are metric functions; Manhattan distance, Minkowski distance and Hamming distance. For non-numeric datasets, special distance functions are proposed. For example, edit distance is a well-known distance measure for text attributes.

**Similarity measure**

An alternative concept to that of the distance is the similarity function \( s(x_i; x_j) \) that compares the two vectors \( x_i \) and \( x_j \) (Duda *et al.*, 2012). This function should be symmetrical (namely \( s(x_i; x_j) = s(x_j; x_i) \)) and have a large value when \( x_i \) and \( x_j \) are somehow “similar” and constitute the
largest value for identical vectors. Jaccard index, Cosine Similarity, Dice Coefficient and Pearson Correlation Measure are also popular distance measures. Although the measures or criteria is considered as important factor for grouping the points in clusters, the components of the clusters are also considered as a important factor for grouping the data points. The components are explained clearly in the following section,

5.3. **Fundamental components used for clustering**

Components such as cluster count, inter and intra-cluster communication, etc. are considered as important factors for the forming the cluster.

**Number of clusters (cluster count)**

In recent probabilistic and randomized grouping algorithms the CH selection and configuration process direct in nature to variable amount of groups. In certain published methods, nevertheless, the set of CHs are fixed in prior and thus the numbers of clusters are predetermined.

**Intra-cluster communication**

In certain primary clustering methods, the communication among a sensor and its selected CH is supposed to assist (one-hop communication). In fact, multi-hop intra-cluster communication is frequently (nowadays) necessary, i.e., in the case of the transmission range of the sensor nodes being limited or the amount of sensor nodes is extremely huge and the amount of CHs is unchanging.

**Nodes and CH mobility**

If assumed the motionless sensor nodes and motionless CHs, are usually led to constant groups with ease intra-cluster and inter-cluster network administration. Conversely, if the CHs or the nodes themselves are presumed to be in a mobile state, the cluster membership for every node must
dynamically vary, thereby forcing clusters to develop over time and hence probably requires continuous maintenance.

**Nodes types and roles**

In some proposed network replica (i.e., heterogeneous environments) the CHs are assumed to be equipped with significantly more computational and communication resources rather than others. In most of the typical network models (i.e., homogeneous environments) all nodes possess similar capabilities, and just subsets of the deployed sensors are designated as CHs.

**Cluster formation methodology:**

In the approaches recently, when CHs are quite usual sensors nodes and time effectiveness is an initial design principle, grouping is being carried out in a distributed way without harmonization. In some earlier methods, a centralized (or hybrid) method is go behind one or more number of coordinator nodes is useful in partitioning the entire network off-line and thus perform the management of the group association.

**Cluster-head selection:**

The leader nodes of the clusters (CHs) in some algorithms proposed (chiefly for heterogeneous environments) could be pre-assigned. In many cases, however (i.e., in the case of homogeneous environments), the CHs are selected from the set of nodes deployed either in a probabilistic or entirely random manner or on the basis of few more criteria that are exact (residual energy, connectivity, etc.).

**Algorithm complexity**

In recent algorithms, the fast annihilation of the protocol carried out is one amongst the significant design objectives. In a similar manner, the time complication or convergence rate of the majority of the cluster formation
processes that are proposed recently is constant (or it is merely based on the amount of CHs or the number of hops). In few of the earlier protocols, however, the complexity time has been permitted to be dependent on the total number of sensors present in the network, concentrating on other criteria first.

**Multiple levels**

In several existing approaches the concept of a multi-level cluster hierarchy is presented to accomplish even better energy distribution and also total energy consumption (rather than making use of just one cluster level). The enhancements rendered by multi-level clustering are to be analyzed further, particularly when this work possesses very big networks, and inter-CH communication efficiency is of primary importance.

**Overlapping**

Many protocols yield high importance on the aspect of node overlapping within the different clusters (either for the cause of improved routing effectiveness or for fast cluster formation protocol implementation or else for dissimilar reasons). Numerous recognized protocols, nevertheless, still endeavor to have least amount of overlap only, else do not offer support for overlapping at all.

**Scalability**

The ability of the algorithm to perform better with a huge number of data objects (tuples).

**Find arbitrary-shaped clusters**

Different types of algorithms will be biased toward getting to know various kinds of cluster structures/shapes and it is not an easy job always to decide the shape or the respective bias. Particularly in the presence of categorical attributes, it may not be relevant to talk about cluster structures.
Minimal requirements for input parameters

Several clustering algorithms need few user-defined parameters, like the number of clusters, for analyzing the data. But, in the case of large data sets and higher dimensionalities, it is desirable that a method requires only limited guidance from the user, in order to avoid biasing the result.

Handling of noise

Clustering algorithms must be capable of handling deviations, for the purpose of improving cluster quality. Deviations are described as data objects that take off from the typically accepted norms of behavior and are also known as outliers. Deviation detection is considered as a separate problem.

Insensitivity to the order of input records

The same data set, if presented to particular algorithms in diverse orders, might lead to dramatically different clusterings. The order of input mostly affects algorithms that perform only single scan on the data set, thus resulting in locally optimal solutions at each step. Thus, it is important that algorithms have to be insensitive towards the order of input.

High dimensionality of data

The number of attributes/dimensions in several data sets is huge, and many clustering algorithms can generate meaningful results only when the number of dimensions is small (e.g., eight to ten).

5.4. Different Types of Clusters

Clustering aims to generate useful sets of objects (clusters), where the usefulness is defined by the targets of the data analysis which is explained by Halkidi et al., (2002). The different types of clusters in
Well-Separated

A cluster refers to a set of objects in which every object is closer (or nearly the same) to every other object in the cluster rather than any object not present in the cluster. Often a threshold is utilized to indicate that all the objects in a cluster must be adequately close (or same) as one another. This idealistic definition of a cluster is met only if the data comprises of natural clusters that are quite distant from each other. The distance between any two points in various groups is greater than the distance between any two points inside a group. Well-separated clusters do not require being globular, but may have any shape.

Prototype-Based

A cluster is a set of objects in which every object is closer (with more similarity) to the prototype that is used for defining the cluster rather than to the prototype of any other cluster. For data with contiguous attributes, the prototype of a cluster is frequently a centroid, i.e., the average (mean) of all the points in the cluster. If a centroid is not meaningful, like when the data contains categorical attributes, the prototype is mostly a medoid, i.e., the typical representative point of a cluster. For several kinds of data, the prototype can be represented as the most central point, and in such situations, this work commonly refers to prototype based clusters as center-based clusters.

Graph-Based

In case of the data being represented in the form of a graph, where the nodes are objects and the links denote connections between objects, then a cluster can be defined as a linked component; i.e., a set of objects which are connected to one another, but those which have no connection to objects that are outside the group. Significant instances of graph-based clusters are contiguity-based clusters, where two objects are connected only when they lie
within a particular distance from each other. This indicates that every object in a contiguity-based cluster is near to some other object in the cluster rather than to any other point in another different cluster.

**Other types of clusters**

Other kinds of graph-based clusters also exist. One kind of this approach defines a cluster in the form of a clique; i.e., a set of nodes in a graph that are fully connected to each other. In specific, if connections between objects are added in the order of their distance from one another, a cluster is generated when a set of objects makes a clique. Just like prototype-based clusters, such clusters try being globular.

Density-Based cluster refers to a dense region of objects which is surrounded by a region of low density.

Shared-Property (Conceptual Clusters) More generally, this work can be used to define a cluster as a group of objects sharing some property. This definition includes all the earlier definitions of a cluster; e.g., objects in a center-based cluster share the property such that they are all closer to the same centroid or medoid. Still, the shared-property approach also contains new kinds of clusters.

**5.5. Benefits of clustering process**

Clustering has many advantages including the reduction of the signaling overhead, abstracting the network topology into a simple and yet a more stable version such that local variations do not trigger the requirement for network-wide updates. Cluster-based scheme presents a virtual backbone for communication where the cluster head plays as the local coordinator of the cluster. In addition, retaining lesser number of nodes in the network backbone effectively enhances the efficiency of the network functions like broadcasting, multicasting, routing, etc. In cluster-based scheme, cluster head collects data from the member nodes and behaves as a fusion center. Data fusion excludes the redundant data during the process data aggregation and eliminates
overhead in data communication. Additionally, clustering renders effective load management and thus helps in increasing the network lifetime. Cluster head may allocate individual tasks to the member nodes for the purpose of load balancing.

These are major reasons for choosing the clustering approach for grouping the nodes before spectrum sensing in proposed framework. The proposed framework is explained in the following section detailed manner.

5.6. Proposed methodology

Conventional sensing methods usually relate to sensing the spectrum in three dimensions such as time, geographic area and frequency. However, there are other dimensions that need to be explored further for spectrum opportunity. It is possible to have a mismatch between the rules in existing spectrum sensing technique. As a result, these types of signals constitute a major problem in sensing the spectrum. In this research, a novel spectrum sensing is proposed in order to overcome the problems in existing spectrum sensing technique. The proposed spectrum sensing technique consists of spectrum segmentation, cluster formation of CRN spectrum sensing with malicious detection. The overall flow of the proposed framework is explained in Figure 5.2.

Figure 5.2 Overall proposed block diagram of secure spectrum sensing

5.6.1. Spectrum Segmentation

Spectrum segmentation is observed as a primary step for identifying the subbands which are in use at a specific time when a large portion of the
spectrum is seen. The observed band is analyzed to find the boundaries of the different subbands; improved histogram based on fuzzy is used for spectrum segmentation which is explained by Dong-Chan and Yong-Hwan (2009). Before that power, spectral density value is calculated for received signal before identifying the boundaries of the band in that signal. Let \([f_0, f_N]\) be the observed frequency range of the radio spectrum. The segmentation process has to estimate \(f_0, f_1, f_2, \ldots, f_N\) the boundaries of the \(N\) frequency intervals. The Figure 5.3 shows the block diagram of the spectrum segmentation process.

**Figure 5.3 Block Diagram of Spectrum Segmentation**

**Step 1:** Calculate the power spectral density value for each and every signal using periodogram function.

**Step 2:** The histogram of the smoothed PSD value (\(F(l)\) values) is calculated first. The local maxima of the histogram, whose values exceed a certain threshold \(m\), are searched. Therefore, the threshold on the maxima detection, \(m\), depends on the minimum bandwidth considered for the sub-bands. Let these maxima be called \(M_1, \ldots, M_k\). Each interval is the distance between two successive local maxima. Then fuzzifies a newly-generated factor from the multiplication of two factors the interval and frequency of signals in the interval. The values are proportional to the sub-band widths.
**Step 3:** Called the PSD level $f_i$ corresponding to the center of the histogram bin, whose occurrence is $M_i$, the PSD segments whose values lie in a range $[f_i - \delta, f_i + \delta]$ are identified, and a new version of the PSD is generated where these segments are rectified to the value $f_i$. The tolerance $\delta$ depends on the variance of the PSD estimate.

**Step 4:** The slope of the rectified version of the PSD between two segments is analyzed to detect a boundary. In particular, the boundary is located where a minimum of the PSD between two segments is found. A subband is found only if the corresponding bandwidth is greater than $l$. The value of $l$ represents the minimum sub-bandwidth. Therefore, it should be chosen starting from the knowledge of the minimum bandwidth of primary users in the observed frequency spectrum.

### 5.6.2. Cluster-Based Spectrum Sensing

Emergent of an efficient Spectrum Sensing (SS) method in Cognitive Radio (CR) is essential that is observed as a consistent system for enhancing the spectrum exploitation. The arrangement of cooperative sensing consists of the PUs, assisting CR users as well as an FC, all the components of cooperative sensing as mentioned in Figure 5.4. Each Cognitive User (CU) performs a spectrum sensing process, which is referred to as local spectrum sensing in the distributed scenario for the detection of the Primary User (PU) signal. Before the CU starts sensing the spectrum value CU’s are grouped as clusters. In case of a CR networks, the location of Secondary User (SU) or CU is randomly distributed. Therefore, some SUs may suffer deep fading while others may not fade. On the other hand, some users may locate near to each other, which experience the same path fading and is supposed to have the same SNR. Therefore CR network is organized into multiple clusters based on the geographical position.
The SNR for primary signal is calculated using the formula is,

\[ SNR = \frac{P_i(d)}{N_i B} \]  

(5.1)

**Figure 5.4 Cluster-based cooperative spectrum sensing**

Where \( P_i \) is a primary signal, \( N_i \) represents the noise of the band and \( B \) is bandwidth. Here, this proposed work suggests a cluster header selection on the basis of the sensing data reliability. Each SU makes use of samples in the sensing interval to do spectrum sensing employing the HANFIS detection technique. The proposed cluster formation process is explained in the Figure 5.5,

**Figure 5.5 Process cluster formation and cluster head selection**
5.6.3. **Spectrum Sensing**

A new technique called as Hybrid Adaptive Neuro-Fuzzy Inference System Model (HANFIS) is introduced in order to identify spectrum holes. The proposed method can predict the channel status whether occupied or unoccupied is designed for spectrum sensing. The power spectral density, capacity over subband, bandwidth efficiency is given as an input to the HANFIS to predict the state of the subband. Spectrum sensing implies the detection of white spaces in the band which is explained by Mitola (2000).

Local spectrum sense at the \( i \)th CU is characteristically a binary hypothesis testing crisis:

\[
\begin{cases}
H_0: x_i(t) = n_i(t) \\
H_1: x_i(t) = h_{ls}(t) + n_i(t)
\end{cases}
\]  

(5.2)

where \( H_0 \) and \( H_1 \rightarrow \) hypotheses of nonexistence and existence of the PU signal correspondingly, \( x_i(t) \rightarrow \) received information at \( CU_i \), \( h_i \) signifies the gain of the channel among the PU and the \( CU_i \), \( s(t) \rightarrow \) signal that is transmitted from the initial user, and \( n(t) \rightarrow \) additive white Gaussian noise. Furthermore, channels with respect to diverse CUs are forecasted to be sovereign, and in addition, all the CUs and PUs share a general spectrum allocation.

Spectrum sensing algorithms may fall into mistakes in practice, which can be classified into miss detections and false alarms. Miss detection occurs when a primary signal is present in the sensed band and the spectrum sensing algorithm selects hypothesis \( H_0 \), which may result in harmful interference to primary users. On the other hand, a false alarm occurs when the sensed spectrum band is idle and the spectrum sensing algorithm selects hypothesis \( H_1 \), which results in missed transmission opportunities and therefore in a lower spectrum utilization. Based on these definitions.
The performance of any spectrum sensing algorithm can be summarized by means of two probabilities: the probability of miss detection \( P_{md} = P(H_0/H_1) \), or its complementary probability of detection \( P_d = P(H_1/H_0) = 1 - P_{md} \) and the probability of false alarm \( P_{fa} = P(H_1/H_0) \). Large \( P_d \) and low \( P_{fa} \) values would be desirable. Nevertheless, there exists a trade-off between \( P_d \) and \( P_{fa} \) meaning that improving one of the performance metrics implies degrading the other one. When the signal is received from the transmitter then the local observation of the \( i^{th} \) user is obtained by equation (5.3),

\[
S_{LOi} = \sum_{n=1}^{S} |x_i(n)|^2
\]  

(5.3)

where \( S \) denotes the number of samples and equals \( 2TB \), and \( T \) and \( B \) are the respective sensing time and bandwidth.

**Hybrid ANFIS**

Adaptive Neuro-Fuzzy Inference System (ANFIS) was first introduced by Jang (1993) & Jang et al., (1997). ANFIS can be easily implemented for a given input/output task, and it is applied to spectrum sensing for identifying the spectrum holes. This means that the ANFIS model is an integration of the ANN and FIS tools into a compound, which again means that there are no boundaries to distinguish the corresponding features of ANN and FIS.

\[
if x_1 \in A_1, x_2 \in A_2, \ldots, x_n \in A_n \text{Then } y = k_0 + k_1 x_1 + \ldots + k_n x_n
\]  

(5.4)

where \( x_1, x_2, \ldots, x_n \) are considered as input signals \( A_1, A_2, \ldots, A_n \) are fuzzy sets and \( y \) is the output variable that can be found in such a type of fuzzy rule. The output variable is obtained as first order polynomial on input variables.
Description of the Method

There are six layers in an ANFIS model which include one input layer, four hidden layers, and one output layer. Each layer is assigned to perform a specific task to transmit the signals. Such an ANFIS model is illustrated in Figure 5.6.

![Figure 5.6 A typical ANFIS model with two inputs and one output](image)

The first layer, i.e., the input layer of the ANFIS model acts as the input layer. Neurons in this layer merely transmit the received input (crisp) signals towards the next layer. Namely,

\[
x_i^1 - y_i^1
\]

(5.5)

where \(x_i^1\) refers the input signal and \(y_i^1\) refers the output signal of neuron \(i\) in the first layer.

The fuzzification layer forms the second layer of the ANFIS model. Neurons in this layer denote the antecedent fuzzy sets of fuzzy rules. A fuzzification neuron at this juncture gets an input signal and determines channel capacity which is used to analyze the transmission rate of the overall channel. The formula used for calculating the channel capacity of the input signal is,
\[ C = B \log_2 (1 + SNR) \]  \hspace{1cm} (5.6)

\[ y_i^2 = f(C(x_i^2)) \]  \hspace{1cm} (5.7)

where \( f \) represents the activation function of neuron \( i \), and is fixed with a specific membership function.

The fuzzy rule layer is the third layer, i.e. the second hidden layer. Every neuron in this layer receives signals only from the fuzzification neurons that are engaged in the antecedents of the fuzzy rule it denotes and calculates the spectral efficiency of the signal.

\[ S_e(x_i) = \frac{B}{\Delta C} \log_2 (1 + SNR) \]  \hspace{1cm} (5.8)

\( \Delta C \) represents ranges of channel capacity. In an ANFIS, the “product” operator is used for evaluating the conjunction of all the neurons. Therefore, those have:

\[ y_i^3 = \prod_{c}^{m} S_e(x_i)c_i \]  \hspace{1cm} (5.9)

where \( S_e(x_i)c_i \) denotes the signal from fuzzification neuron \( c \) in the second layer to neuron \( i \) in the third layer, \( y_i^2 \) represents the output signal of neuron \( i \) in this layer and \( m \) denotes the number of antecedents of the fuzzy rule that the neuron \( i \) represents .

The normalization layer forms the fourth layer. Every neuron in this layer obtains signals from all the rule neurons in the third layer and computes the so-called normalized firing strength of a rule given. This strength value indicates the threshold value of the channel capacity and spectral efficiency which is used to determine the state of the spectrum subband.

The defuzzification layer forms the fifth layer. Every neuron in this layer is connected to the corresponding normalization neuron in the fourth layer and also gets \( x_1, x_2, \ldots, x_n \).
A defuzzification neuron provided the computation of the “weighted consequent value” of a given rule as below:

\[ y_i^5 = x_i^5(k_{i0} + k_{i1}x_1 + k_{i2}x_2 + \cdots + k_{in}x_n) \quad (5.10) \]

where \( x_i^5 \) refers to the input and \( y_i^5 \) refers to the output signal of neuron \( i \) in the fifth layer; and \( k_{i0} + k_{i1} + k_{i2} + \cdots + k_{in} \) represents a set of the resultant parameters of rule \( i \).

The sixth layer, i.e. the output layer corresponds to the summation layer. Only one neuron exists in the layer, which calculated the summation of the outputs of all defuzzification neurons in the fifth layer and as a result generates the overall ANFIS output \( y \) as follows:

\[ y = \sum_{i=1}^{n} x_i \quad (5.11) \]

where \( x_i \) refers to the signal from defuzzification neuron \( i \) in the fifth layer to this summation neuron; and \( n \) denotes the number of defuzzification neurons, such as the number of fuzzy rules in the ANFIS model.

**Training ANFIS Model**

For ANFIS models, the generally employed activation function is the one called the bell-shaped function, defined as:

\[ y = \frac{1}{1 + \left[ \left( x - \frac{s}{r} \right) \right]^t} \quad (5.12) \]

where \( r, s \) and \( t \) denote parameters that control the slope, center and width of the bell-shaped function respectively. Moreover in the training process, these parameters can be defined and accordingly adjusted by the learning algorithm.
In particular, an ANFIS uses a hybrid learning (training) algorithm. This learning algorithm performs the combination of the least-squares estimator and the gradient descent method lastly with Runge-Kutta Learning Method (RKLM). To start with, the initial bell-shaped functions with specific parameters are allocated to each of the fuzzification neuron. The function center of the neurons that are connected to input $x_i$ are determined such that the domain of $x_i$ is divided in a uniform manner, and the function widths and slopes are set such as to permit adequate overlapping(s) of the corresponding functions which is described by Nazzal et al., (2008). When the training process is on, the training dataset is given to the ANFIS in a cyclic way. Each cycle going through all the training examples is known as an epoch. In the case of the ANFIS learning algorithm, each epoch comprises of one forward pass and one backward pass. The aim of the forward pass is the formation and adjustment of the consequent parameters, whereas the backward pass is for adjusting the parameters of the activation functions.

5.6.4. **Cluster Head Selection**

The node that has dependable sensing outcome will take up the cluster header’s responsibilities that comprise the formation and reporting of the cluster’s decision to the FC. For the intention of diminishing the accounting time and bandwidth, it is merely the sensing information of the cluster header, which is also the mainly consistent one, is used in order to figure out the cluster decision. This technique specifies that the verdict of a cluster is made based on the selective combination technique. The FC will merge every cluster decision for the purpose of making a final decision and then broadcast the final sensing decision to the entire network. This work suggests a cluster header selection which is based on the reliability of sensing information. For each and every sensing interval, the CU having the unswerving sensing information in a cluster is selected as the cluster header (CH). The sensing information of the CR is deliberated by the probability of
detection $P_d$ which is again a measure of the interference to the PU and also the probability of false alarm $P_F$ which fixes the upper bound on the utilization of the spectrum. The detection and false alarm probabilities of the $i^{th}$ user are respectively provided, as Probability of detection is,

$$P_{d,i} = P(y_i > \lambda_i | H_1) = Q\left(\frac{\lambda_i - N(\gamma_i + 1)\sigma_u^2}{\sigma_u^2 \sqrt{2N(2\gamma_i + 1)}}\right)$$  \hspace{1cm} (5.13)

where $\lambda_i$ is local channel capacity threshold, $\gamma_i$ is the Signal to noise ratio and $\sigma_u^2$ is the variance. $N$ is the number of sample received signals.

Probability of false alarm

$$P_{f,i} = P(y_i < \lambda_i | H_0) = \left(\frac{\lambda_i - N\sigma_u^2}{\sigma_u^2 \sqrt{2N}}\right)$$  \hspace{1cm} (5.14)

where $\lambda_i$ is local channel capacity threshold, and $\sigma_u^2$ is the variance. $N$ is the number of sample received signals.

Based on the sensed data results, reliability of each CRU is calculated, and the reliability value is used for selecting the cluster head. The other CRN are joined as member to the cluster based on geographical position. The header in each of the cluster is not pre-determined but selected dynamically for every sensing interval according to the quality of the sensing data at every CU.

5.6.5. **Attacks Detection**

The performance of spectrum sensing can be severely damaged by malicious secondary users. The malicious secondary user is defined by using certain parameters in the network such as attack strength and attack probability. To address this problem the suspicious level of the nodes are calculated in the cluster and suspicious value is stored in the cluster head as a table. Figure 5.7 illustrates the malicious user detection flow of the proposed method.
Suspicious level calculation

Assume that the malicious user is present in the network. This work defines secondary users as,

\[ S_i(t) \triangleq P(T_i = M|F_t) \tag{5.15} \]

as the suspicious level of node \( i \) and \( j \) at time \( t \), where \( T_n(= H \text{ or } M) \) is the type of node and \( F_t \) represents all observations from time slot 1 to time slot \( t \). The suspicious value calculation is described by Wang et al. (2009). The formula used for calculating the suspicious value is,

\[ S_i(t) = \frac{\prod_{\tau=1}^{t} \theta_i(\tau)}{\sum_{j=1}^{N} \prod_{\tau=1}^{t} \theta_j(\tau)} \tag{5.16} \]

where \( \tau \) is a time slot, \( N \) is number of secondary users present in the cluster and \( \theta_i(\tau) \) is a probability of reports at time slot \( t \) conditioned that node \( n \) is malicious.

\[ \theta_i(\tau) = P_{d,i} + P_{f,i} \tag{5.17} \]

The suspicious values of all the users in the cluster are calculated, and the values are updated in cluster head. After the spectrum sensing, the node suspicious value are calculated if the suspicious value is below the threshold then node is honest node whereas the value is beyond the threshold.
then node is malicious. Once the node is identified as malicious node, then the cluster head forward the message to the cluster members to isolate the suspicious node.

5.7. Experimental results

In this section, experimentation is carried out to evaluate the performance of the proposed secure spectrum sensing technique, in comparison with conventional techniques and the results are provided. It is assumed that the minimum distance between the secondary users and the primary user is 1000m, and the maximum distance is 2000m. For the local spectrum sensing, the bandwidth-time product (Ghasemi and Sousa 2005); (Ghasemi and Sousa 2007) is $m = 5$. The transmission power of the primary user is 200mW. The noise level $\sigma^2$ is -110dBm.

![Figure 5.8 Comparison of LMMSE vs. noise power](image)

The Signal-to-Noise Ratio (SNR) of the individual secondary user depends on its location and Rayleigh fading is assumed. The parameter used to
measure the performance of the proposed technique is linear MMSE value. It should be noticed that the proposed method simulation is not conducted over a physical network model since this work does not rely on any physical layer setting. In a cognitive radio system, each SU has a detection probability $P_{d,i}$ and a false alarm probability $P_{f,i}$ on a primary channel.

In Figure 5.8, the Linear Minimum Mean-Square Errors (LMMSEs) of the proposed spectrum sensing algorithm is contrast with the existing algorithms like spectrum sensing supported NN and HSMM sourced spectrum sensing versus the noise variance $\sigma_n^2$. The sensing unit is modeled to have a detection probability of $P_d = 0.6$ and a false-alarm probability of $P_f = 0.2$. It is observed that the proposed algorithm achieves the lowest LMMSEs whereas other algorithms exhibit larger LMMSEs. In addition, as the noise variance increases, the LMMSE’s increase and the performance of the estimators of the proposed approach and existing approaches became close to each other. Table 5.1 shows the values of the comparison of LMMSE vs noise power for proposed and existing algorithms.

<table>
<thead>
<tr>
<th>Noise Power</th>
<th>HSMM based spectrum sensing</th>
<th>FNN based spectrum sensing</th>
<th>Hybrid Secure Spectrum sensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.480</td>
<td>0.4650</td>
<td>0.3610</td>
</tr>
<tr>
<td>0.4</td>
<td>0.506</td>
<td>0.4932</td>
<td>0.4634</td>
</tr>
<tr>
<td>0.6</td>
<td>0.548</td>
<td>0.5351</td>
<td>0.5183</td>
</tr>
<tr>
<td>0.8</td>
<td>0.593</td>
<td>0.5638</td>
<td>0.5429</td>
</tr>
<tr>
<td>1.0</td>
<td>0.613</td>
<td>0.5943</td>
<td>0.5618</td>
</tr>
</tbody>
</table>

Table 5.1 Comparison of LMMSE vs. noise power
Detection probability

Figure 5.9 Comparison of LMMSE vs. detection probability

Table 5.2 Comparison of LMMSE vs. detection probability

<table>
<thead>
<tr>
<th>Detection probability</th>
<th>HSMM based spectrum sensing</th>
<th>FNN based spectrum sensing</th>
<th>Hybrid Secure Spectrum sensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.4130</td>
<td>0.3932</td>
<td>0.3735</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4386</td>
<td>0.4102</td>
<td>0.3796</td>
</tr>
<tr>
<td>0.7</td>
<td>0.4432</td>
<td>0.4338</td>
<td>0.3813</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4516</td>
<td>0.4513</td>
<td>0.3918</td>
</tr>
<tr>
<td>0.9</td>
<td>0.4594</td>
<td>0.4738</td>
<td>0.4035</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5612</td>
<td>0.4997</td>
<td>0.4106</td>
</tr>
</tbody>
</table>
In Figure 5.9, the Linear Minimum Mean-Square Errors (LMMSEs) of the proposed spectrum sensing algorithm is measure with the existing algorithms alike spectrum sensing supported on HSMM and NN supported spectrum sensing against detection probability. In the Figure 5.9, HSMM provides highest LMMSE since some of the parameters such as channel capacity, SNR of primary signal and bandwidth efficiency is not considered during spectrum sensing. It is observed that the proposed algorithm achieves the lowest LMMSEs whereas other algorithms have a poor performance, as expected. Table 5.2 shows the values of the comparison of LMMSE vs. detection probability for proposed and existing algorithms.

**False alarm probability**

![Figure 5.10 Comparison of LMMSE vs. false alarm probability](image-url)

In Figure 5.10, the LMMSEs values of the proposed algorithm is compared with the existing algorithm such as spectrum sensing based on HSMM and NN-based spectrum sensing versus the false-alarm probability for
a detection probability of $P_d = 0.6$ and a noise variance of $\sigma_n^2 = 0.2$. It is observed that the LMMSE’s increase as the false-alarm probability increases. This is mainly because the power of the pilot symbol is reduced ($P_{t1} = 0.1$ is employed) in the presence of a false alarm; that is, when the channel sensing unit decides that the primary users are present in the system when in fact they are not. In the Figure 5.10, HSMM provides highest LMMSE since some of the parameters such as channel capacity, SNR of primary signal and bandwidth efficiency is not considered during spectrum sensing. Table 5.3 shows the experimental values of the proposed algorithm and existing algorithms.

**Table 5.3 Comparison of LMMSE vs. false alarm probability**

<table>
<thead>
<tr>
<th>False alarm probability</th>
<th>HSMM based spectrum sensing</th>
<th>FNN based spectrum sensing</th>
<th>Hybrid Secure Spectrum sensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.3689</td>
<td>0.3231</td>
<td>0.3024</td>
</tr>
<tr>
<td>0.1</td>
<td>0.3838</td>
<td>0.3618</td>
<td>0.3413</td>
</tr>
<tr>
<td>0.2</td>
<td>0.4037</td>
<td>0.3916</td>
<td>0.3618</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4514</td>
<td>0.4276</td>
<td>0.3917</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4836</td>
<td>0.4518</td>
<td>0.4218</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5137</td>
<td>0.4812</td>
<td>0.4467</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5376</td>
<td>0.5017</td>
<td>0.4746</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5598</td>
<td>0.5218</td>
<td>0.4935</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5949</td>
<td>0.5651</td>
<td>0.5108</td>
</tr>
<tr>
<td>0.9</td>
<td>0.6269</td>
<td>0.5912</td>
<td>0.5230</td>
</tr>
</tbody>
</table>
Power Spectral Density (PSD) of the received signal and a simplified function of frequency are minimized. The best fitting simplified function is then used to estimate the subband boundaries.

In the previous segmentation process, segmentation can only be done by checking the N boundaries $f_0, f_1, ..., f_N$ of N frequency intervals. Here $f_0$ and $f_N$ is minimum and maximum frequency respectively. Hence $f_0 = 100\text{Hz}$, $f_N=500\text{Hz}$. However in the proposed methodology namely PSD $f_0$ boundary itself enough for the efficient segmentation. These boundaries would attempt to identify the sub-bands, so that the active transmission can be done. Thus segmentation in the boundaries has to be accurate result for achieving active transmission. This segmentation can give qualitative information about each detection stage by using which different operating decision can be taken. May be in future work we simulate experiments under varied frequency ranges under a different number of node ranges.

![Power Spectral Density](image)

**Figure 5.11 (a) PSDs of the optimal scheduling considered during the experimental validation phase**

The experimental validation stage has been organized by creating four methods such as optimal scheduling, Hidden Semi-Markov Model (HSMM), Fuzzy Neural Network (FNN) and Hybrid Adaptive Neuro Inference System
(ANFIS), in which several signals are located on different sub-bands. Two of them represent under various configurations that can be found in certain telecommunication bands. In particular, the optimal scheduling scenario (Figure. 5.11 (a)) includes different power density levels and symbol frequencies. The second HSMM (Figure. 5.11(b)) represents a frequency band, including signals. In this case, the signals located in the different subbands have different power density levels, but the same symbol period, since the spectrum segmentation is not performed in this schema.

Figure 5.11(b) PSDs of the Hidden Semi-Markov Model (HSMM) considered during the experimental validation phase

A third Fuzzy Neural Network (FNN) scenario has been added to verify the limits of the methods in critical conditions (Figure 5.11 (c)). It has been observed, in fact, that the proposed method can fail when signals with high differences in the PSD slopes are present within the observed spectrum. Therefore, in this third scenario, two signals have been provided. The former has a wide band and a high roll-off factor such that the PSD slope is very smooth; the latter is a narrowband signal with a low roll-off factor, thus having a sharp slope. Such a scenario has been repeated several times, by progressively reducing the bandwidth of the narrowband signal thus increasing spectrum sensing results. In Figure 5.11 (c), one of the PSDs of such Hybrid
Adaptive Neuro Inference System (ANFIS) scenario is reported under malicious node identified results. So, this ANFIS model has also been repeated several times, by progressively reducing the bandwidth of the narrowband signal thus increasing spectrum sensing results and identification of malicious nodes.

Figure 5.11 (c) PSDs of the Fuzzy Neural Network (FNN) considered during the experimental validation phase

Figure 5.11 (d) PSDs of the Adaptive Neuro-Fuzzy Interference System (ANFIS) considered during the experimental validation phase
Figure 5.12 False alarm probability of $\alpha$ when the population size increases to 1,000, nodes

Figures 5.12 show the False alarm probability of density of malicious nodes $\alpha$, the ratio of malicious versus total number of nodes as 1000. The results confirm the behavior discussed earlier; we see a clear separation of the two classes only when the malicious nodes and non-malicious nodes. When the density of malicious nodes approaches is 0.7341 for 0.6 density values. It shows that the false alarm detection probability of the proposed schema is high value when compare to other values. The false alarm detection probability of the other schemas such as optimal scheduling, HSMM and FNN are 0.302, 0.3361 and 0.5581 respectively which is very less when compare to proposed schema for the density value of 0.6, the values are tabulated in table.
Table 5.4. False alarm probability of $\alpha$ when the population size increases to 1,000, nodes

<table>
<thead>
<tr>
<th>Density value ($\alpha$)</th>
<th>False alarm probability</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>optimal scheduling</td>
<td>HSMM</td>
<td>FNN</td>
<td>HANFIS</td>
</tr>
<tr>
<td>0.20</td>
<td>0.150</td>
<td>0.190</td>
<td>0.420</td>
<td>0.620</td>
</tr>
<tr>
<td>0.25</td>
<td>0.180</td>
<td>0.210</td>
<td>0.435</td>
<td>0.638</td>
</tr>
<tr>
<td>0.30</td>
<td>0.210</td>
<td>0.230</td>
<td>0.450</td>
<td>0.645</td>
</tr>
<tr>
<td>0.35</td>
<td>0.230</td>
<td>0.254</td>
<td>0.486</td>
<td>0.658</td>
</tr>
<tr>
<td>0.40</td>
<td>0.245</td>
<td>0.261</td>
<td>0.492</td>
<td>0.663</td>
</tr>
<tr>
<td>0.45</td>
<td>0.262</td>
<td>0.281</td>
<td>0.510</td>
<td>0.681</td>
</tr>
<tr>
<td>0.50</td>
<td>0.271</td>
<td>0.310</td>
<td>0.536</td>
<td>0.710</td>
</tr>
<tr>
<td>0.55</td>
<td>0.291</td>
<td>0.321</td>
<td>0.541</td>
<td>0.725</td>
</tr>
<tr>
<td>0.60</td>
<td>0.302</td>
<td>0.336</td>
<td>0.558</td>
<td>0.734</td>
</tr>
</tbody>
</table>

5.8. Summary

This chapter summarizes the methodology of spectrum sensing algorithm for Cognitive Radio Networks (CRNs). The major steps are being carried out in the proposed work in which one is for spectrum segmentation based on Fuzzy histogram, the second one is for clustering of cognitive radio nodes, and the third one is spectrum sensing based on Hybrid Adaptive Neuro-Fuzzy Inference System (HANFIS) classifier for identifying the empty spectrum in the CRN. The proposed HANFISSS enables achieving higher classifying accuracy to enhance the spectrum utilization process by noise uncertainty. From the simulation results of the proposed HANFISSS, it was observed that the value attained the Linear Minimum Mean-Square Error (LMMSE) value attained for the proposed HANFISSS approach, the Linear Minimum Mean-Square Error value attained for 0.5 probability detection is 0.3735 which is 0.0395 and 0.0197 lesser than HSMMSS and FNNSS approaches respectively. Thus the proposed HANFISSS helps to achieve the efficient utilization of spectrum bands in cognitive radio network.
CHAPTER 6
PERFORMANCE EVALUATION OF COGNITIVE RADIO NETWORKS WITH SPECTRUM SENSING SEGMENTATION METHODS

6.1. Introduction

In this section, the numerical results are provided to evaluate the performance achieved by the proposed Adaptive Neuro-Fuzzy Inference system (ANFIS) spectrum sensing segmentation method, in comparison with conventional Continuous Wavelet-Based Transform (CWT) methods and Phase-Field Segmentation (PFS).

Continuous Wavelet Transform (CWT)

Continuous Wavelet Transform (CWT) modulus had been used to handle detecting vacant frequency subbands by picking the local maxima of their first derivatives as discussed in Tian and Giannakis (2006). This method identifies the locations and intensities of vacant (spectrum holes) and occupied frequency bands, by analyzing the irregularities in the PSD. Considering a wide bandwidth in the frequency range \([f_0, f_N]\), i.e., the bandwidth is corresponding to \(B = f_N - f_0\) Hz, consisting of \(N\) different subbands, where the \(n^{th}\) subband \(B_n\) is equivalent to the frequency range \(f_{n-1} < f < f_n\), for \(n = 1 . . . N\).

The following fundamental assumptions for the channel spectrum \(S_r(f)\) is considered as discussed in Tian and Giannakis (2006):

- The frequency boundaries \(f_0\) and \(f_N\) are known to the CR.
- The number of total frequency subbands \(N\) and the positions of the spectrum boundaries \(f_1, f_2, . . . f_{N-1}\) are not known to the CR. These data remain unaltered within a time burst, but may differ from burst to burst in the occurrence of slow fading.
• The PSD within each subband $B_n$ is smooth and almost flat, however displays discontinuities from its neighbor subbands.

• The ambient noise is additive and white, with zero mean and two-sided PSD $S_z(f) = N_0/2$, $\forall f$.

Estimates of frequency boundaries of consecutive subbands $\{B_n\}$ are attained through an appropriate method. For instance, in Tian and Giannakis (2006), these boundaries are identified through the local maxima of the first derivatives of the Continuous wavelet transform modulus. Then, the CR has to compute the PSD levels $\alpha_n^2$. This is attained by calculating the average PSD within the $n^{th}$ subband $B_n$ for $n = 1 \ldots N$. The computed value is compared to an energy threshold to classify the subband as vacant or occupied. However, considering the practical scenarios, this Wavelet-based edge detection approaches failed to offer an accurate spectrum sensing as discussed in Tian and Giannakis (2006).

**Phase-Field Segmentation (PFS)**

Eslami and Sadough (2013) proposed Phase-Field Segmentation (PFS) method for detecting vacant frequency subbands for opportunistic cognitive radio which is the other widely used conventional approach. This is equivalent to certain edges in the PSD of a wideband channel, i.e., switching from an engaged band to an empty band or vice versa. Thus, the power within two edges can be computed, and the CR can distinguish whether a frequency band is empty or occupied if the edges are detected. But, edge detection approach is based on PFS, which is advantageous over the wavelet-based approaches. Alternatively, Mumford–Shah segmentation method (Mumford and Shah 1989) could also be utilized for detecting the edges present in an image wherein the edges and segments in a given image are extracted from the minimization of a specific function. Phase-field approach (Ambrosio and Tortelli, 1992), can be used as an approximation approach for solving the
optimization problem in the Mumford–Shah segmentation approach. This approach is observed to provide lower complexity due to the influence of a numerical solution for identifying the PSD’s edges along with more flexibility to adapt a particular scenario featured by the unknown PSD.

This chapter clearly provides an evaluation of the proposed ANFIS based edge detection method to possible errors on the value of different parameters (edge threshold values, PSD sharpness). The performance of the proposed ANFIS approach is compared with the above discussed conventional PFS and CWT techniques.

6.2. Energy detection based spectrum sensing

In recent times, owing to its small complexity and computational cost, energy detection based spectrum sensing is the most familiar spectrum sensing technique. It is carried out by evaluating the received energy of the signal in opposition to a predetermined energy detection threshold to decide the existence or nonexistence of the user in the frequency band of attention. Energy detector has a band-pass filter which restricts the bandwidth of the received signal to the frequency band of interest, a square law device which squares each term of the received signal and a summation device which sums the entire squared values to figure out the energy. The energy is computed using equation (6.1),

$$ E = \sum_{n=0}^{\infty} |x(n)|^2 $$ (6.1)

The energy is now compared against a threshold for inspecting which hypothesis becomes true,

$$ E < \lambda \Rightarrow H_0 $$ (6.2)

$$ E > \lambda \Rightarrow H_1 $$ (6.3)
The probability of detection \((P_d)\) and probability of false alarm \((P_{fa})\) are given by

\[
P_d = Q_m \left( \sqrt{2\gamma, \sqrt{\lambda}} \right) \tag{6.4}
\]

\[
P_{fa} = \frac{\Gamma \left( m, \frac{\lambda}{2} \right)}{\Gamma(m)} \tag{6.5}
\]

where \(\Gamma(m)\) represents the complete gamma function and \(\Gamma \left( m, \frac{\lambda}{2} \right)\) indicates the incomplete gamma function, \(\gamma\) and \(\lambda\) indicates SNR and detection threshold correspondingly. The performance of energy detector dependent sensing is restricted in case of the two common assumptions do not hold. The noise possibly will not be stationary, and its variance might not be recognized. Other complications with the energy detector comprise baseband filter effects and spurious tones.

Figure 6.1 Probability of miss-detection against the energy threshold value
From the figure 6.1, it is observed that under appropriate edge threshold values, the performance of the three spectrum segmentation sensing methods are very close. However, when the edge threshold value is not accurately selected, at this point it observed a degradation of the three spectrum segmentation sensing methods. More precisely, the proposed Adaptive Neuro-Fuzzy Inference system (ANFIS) spectrum sensing segmentation method is less sensitive to inaccurate edge threshold values (for instance, to imperfect noise power estimation) than the conventional Phase-Field Segmentation (PFS) and Continuous Wavelet-Based Transform (CWT) methods. It shows that the probability of miss-detection for the proposed ANFIS spectrum segmentation decreases slightly when energy threshold value increases. For energy threshold value 1, the proposed ANFIS spectrum segmentation method attains 0.04218 probability of miss detection value, whereas probability of miss detection value of the Phase-Field Segmentation (PFS) and Continuous Wavelet-Based Transform (CWT) methods are 0.05698 and 0.07841 values respectively. Hence, the proposed ANFIS spectrum segmentation attains less probability of miss detection when compared to PFS and CWT values such as 0.0148 and 0.03623 respectively.

**Table 6.1 Probability of miss-detection versus the energy threshold value**

<table>
<thead>
<tr>
<th>Energy Threshold</th>
<th>Probability of miss-detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CWT</td>
</tr>
<tr>
<td>0.3</td>
<td>0.750</td>
</tr>
<tr>
<td>0.4</td>
<td>0.635</td>
</tr>
<tr>
<td>0.5</td>
<td>0.425</td>
</tr>
<tr>
<td>0.6</td>
<td>0.214</td>
</tr>
<tr>
<td>0.7</td>
<td>0.145</td>
</tr>
<tr>
<td>0.8</td>
<td>0.098</td>
</tr>
<tr>
<td>0.9</td>
<td>0.081</td>
</tr>
<tr>
<td>1.0</td>
<td>0.078</td>
</tr>
</tbody>
</table>
Figure 6.2 ROC curves of the Phase-Filed Segmentation (PFS), Continuous Wavelet-Transform (CWT), and ANFIS spectrum sensing methods.

In order to show the sensitivity to the choice of the energy threshold value, Figure 6.2 depict the ROC curves for the Phase-Field Segmentation (PFS), Continuous wavelet-Transform (CWT) based method, proposed ANFIS method, respectively. In Figure 6.2, a “good” energy threshold is the value for which almost all energy is found by the adopted spectrum sensing method and conversely, a “bad” energy threshold value is the value for which most of the energy is not detected. Here, “good” energy threshold value is equal to 0.8 and “bad” energy thresholds are respectively equal to 0.4 and 0.6. For probability of false alarm 1, the proposed ANFIS spectrum segmentation method attains 0.06893 probability of miss detection, whereas probability of miss detection of the Phase-Field Segmentation (PFS) and Continuous Wavelet-Based Transform (CWT) methods are 0.08246 and 0.1046 respectively. Hence the proposed ANFIS spectrum segmentation attains lesser probability of miss
detection when compared to PFS and CWT values such as 0.01353 and 0.03567.

Table 6.2 Probability of miss-detection versus the Probability of false alarm

<table>
<thead>
<tr>
<th>Probability of false alarm</th>
<th>Probability of miss-detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CWT</td>
</tr>
<tr>
<td>0.0001</td>
<td>0.6812</td>
</tr>
<tr>
<td>0.001</td>
<td>0.6228</td>
</tr>
<tr>
<td>0.01</td>
<td>0.4631</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1853</td>
</tr>
<tr>
<td>1</td>
<td>0.1046</td>
</tr>
</tbody>
</table>

Figure 6.3 ROC curves of the spectrum sensing segmentation methods at different SNR values

Figure 6.3 shows the ROC curves for the proposed ANFIS and Phase-Field Segmentation (PFS), Continuous Wavelet-Transform (CWT) method at different SNR values of 10 and 8 dB while “good” energy
thresholds are selected. As expected, the spectrum sensing performance becomes worse when the SNR decreases. For probability of false alarm 1 at SNR =10 dB, the proposed ANFIS spectrum segmentation method attains 0.05649 probability of miss detection, whereas probability of miss detection of the Phase-Field Segmentation (PFS) and Continuous Wavelet-Based Transform (CWT) methods are 0.1857 and 0.1385 values respectively. Hence, the proposed ANFIS spectrum segmentation attains lesser probability of miss detection when compared to CWT and PFS values such as 0.12921 and 0.08201 respectively.

Table 6.3 ROC curves of the spectrum sensing segmentation methods at different SNR values

<table>
<thead>
<tr>
<th>Probability of false alarm</th>
<th>Probability of miss-detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR=8 dB</td>
</tr>
<tr>
<td></td>
<td>CWT</td>
</tr>
<tr>
<td>0.0001</td>
<td>0.7125</td>
</tr>
<tr>
<td>0.001</td>
<td>0.6228</td>
</tr>
<tr>
<td>0.01</td>
<td>0.4631</td>
</tr>
<tr>
<td>0.1</td>
<td>0.3281</td>
</tr>
<tr>
<td>1</td>
<td>0.2187</td>
</tr>
</tbody>
</table>

6.3. Summary

In this chapter, the experimental results of the proposed Adaptive Neuro-Fuzzy Inference system (ANFIS) spectrum sensing segmentation method, in comparison with conventional Phase-Field Segmentation (PFS) and Continuous Wavelet-Based Transform (CWT) methods for wideband channels in Cognitive Radio (CR) systems are obtained. The spectrum segmentation
technique, depending on the assessment of the improved histogram of the Power Spectral Density (PSD) in the observed band is formulated for recognizing the subband boundaries, and it is exemplified by a low computational load. Second, the smoothed spectrum can be used for reducing complexity of the estimation of the PSD levels inside each subband, since the median value of the smoothed spectrum can be used instead of integration of the initial noisy PSD over each subband. Adaptive Neuro-Fuzzy Inference system (ANFIS) is employed here for the purpose of identification of the malicious user in the spectrum that might be allocated to the aspiring Secondary User (SU).