LEARNING-BASED SPECTRUM SENSING AND ACCESS FOR COGNITIVE RADIO SYSTEMS

by

Menatalla Diaaeldin Shehabeldin

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Electrical Engineering

Sharjah, United Arab Emirates

June 2015

Approval Signatures

We, the undersigned, approve the Master's Thesis of Menatalla Diaaeldin Shehabeldin.

Thesis Title: Learning-Based Spectrum Sensing and Access for Cognitive Radio Systems

Signature	**Date of Signature** (dd/mm/yyyy)
_____	_____
Dr. Mohamed El-Tarhuni Professor of Electrical Engineering Thesis Advisor	
_____	_____
Dr. Khaled Assaleh Professor of Electrical Engineering Thesis Advisor	
_____	_____
Dr. Mahmoud Ibrahim Visiting Associate Professor of Electrical Engineering Thesis Committee Member	
_____	_____
Dr. Tamer Shanableh Associate Professor of Computer Science Thesis Committee Member	
_____	_____
Dr. Nasser Qaddoumi Head of Department of Electrical Engineering	
_____	_____
Dr. Mohamed El-Tarhuni Associate Dean, College of Engineering	
_____	_____
Dr. Leland Blank Dean, College of Engineering	
_____	_____
Dr. Khaled Assaleh Interim Vice Provost for Research and Graduate Studies	

Acknowledgements

I would like to thank all those who have contributed in one way or another to the final presentation of this thesis. First, I would like to thank my advisors, Dr. Mohamed El-Tarhuni and Dr. Khaled Assaleh, who provided continuous support and guidance throughout the research process. Second, I would like to thank all the professors and lab instructors I worked with as a teaching assistant during the master's program for helping me realize that I want to pursue a career in academia. Special thanks to the Department of Electrical Engineering for providing me with the assistantship to continue my studies. I would also like to thank my friends Ayah and Rabiya for making my experience in the program a memorable one. Last but not least, I would like to thank my family for being considerate at the times I got busy with my work on the thesis.

To Mom . . .

Abstract

Spectrum management is one of the most important elements of the overall design of cognitive radio systems. Primary users (PUs), or license holders, should not be affected by the opportunistic use of the spectrum by the secondary users (SUs). Moreover, secondary users, or the non-license holders, should try to maximize their utilization of free channels for better spectrum efficiency. The decision whether to access a channel or not is crucial to both the primary and secondary users. In this thesis, an improved spectrum access algorithm is proposed for cognitive radio systems by modeling the primary user channel usage pattern as a Hidden Markov Model (HMM). The proposed algorithm maximizes the channel utilization without causing significant interference to the primary user by considering access based on the availability of the channel at the current timeslot. The decision on the availability of the channel is investigated using three machine learning techniques, namely HMMs, polynomial classifiers and nonlinear autoregressive with exogenous inputs (NARX) models. Simulation results based on models from real spectrum measurements show that using the conventional HMM-decoding technique leads to high collision probabilities of around 25%. On the other hand, using polynomial classifiers for deciding the availability of the channel enhances the system performance significantly, with collision probabilities less than 1%, while maintaining high utilization probabilities. A thorough investigation of the effect of the order of the polynomial classifier shows that while lower orders reduce the computational complexity of the algorithm, higher orders are more robust to high levels of shadowing. Another approach to mitigate the effect of shadowing is using cooperative spectrum sensing, where multiple SUs send the sensing results to a fusion center, which makes a global decision about the availability of the channel. Results show that the decision based on the scores of the classifiers outperforms majority vote in terms of collision and utilization probabilities.

Search Terms: ***cognitive radio, spectrum management, dynamic spectrum access, spectrum sensing, hidden Markov model, polynomial classifier, nonlinear autoregressive with exogenous inputs model***

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

ANN Artificial Neural Network

AWGN Additive White Gaussian Noise

BIOSS BIOlogically-inspired Spectrum Sharing

BPTT Back-Propagation Through Time

CR Cognitive Radio

CRN Cognitive Radio Network

CRS Cognitive Radio System

CTMC Continuous-Time Markov Chain

DSA Dynamic Spectrum Access

DTMC Discrete-Time Markov Chain

e-BIOSS enhanced BIOlogically-inspired Spectrum Sharing

ED Energy Detection

EM Expectation Maximization

GMM Gaussian Mixture Model

HMM Hidden Markov Model

kNN k-Nearest Neighbors

MDP Markov Decision Process

NARX Nonlinear AutoRegressive with eXogenous inputs model

pdf probability density function

pmf probability mass function

POMDP Partially-Observable Markov Decision Process

PSO Particle Swarm Optimization

PU Primary User

QoS Quality of Service

SNIR Signal-to-Noise and Interference Ratio

SNR Signal-to-Noise Ratio

SU Secondary User

SVM Support Vector Machine

WRAN Wireless Regional Area Network

Chapter 1: Introduction

Cognitive radio networks are thought to be the next generation networks that will overcome the spectrum scarcity problem using dynamic spectrum access techniques. Since the static spectrum allocation currently adopted by regulation authorities has led to a number of spectrum bands being unused at different times of the day, the bands can be utilized by other users who can access the band opportunistically. Research in cognitive radio is currently one of the most active in the field of wireless communications since it provides a promising alternative to the existing fixed spectrum allocation policies.

One of the major challenges in cognitive radio system design is spectrum management. The secondary users (SUs), or the non-license holders, can access the spectrum band only when it is not in use by a primary user. The decision whether or not to access the channel is based on the analysis of the channel properties. Knowing how likely the channel is to be free, what quality of service (QoS) the SU would get, etc., are some of the main problems that are addressed in spectrum management.

In this thesis, the problem of spectrum management is viewed from a machine-learning perspective. The primary user (PU) channel activity is modeled as a hidden Markov model (HMM), and the channel availability is decided by three different techniques, namely HMMs, polynomial classifiers and nonlinear autoregressive with exogenous inputs (NARX) models. Furthermore, a channel access algorithm that improves the utilization of primary user channels is proposed.

1.1. Problem Statement

In a cognitive radio network, the secondary users, or the users who do not have license to the spectrum, are required to vacate the channel once a primary user, or the license holder, needs to use the channel. The PU should not be affected, under any circumstances, by the opportunistic use of the SUs. Simultaneously, the requirements of the SU should also be satisfied, which requires proper spectrum management.

Given a PU and an SU(s), it is required that the SU maximizes the utilization of spectrum holes to increase spectrum efficiency and at the same time, minimizes the

probability of colliding with a PU. This is achieved by an accurate spectrum sensing mechanism along with a proper spectrum access scheme.

1.2. Motivation

Cognitive radio is a promising solution for the spectrum scarcity problem. The current static allocation policies lead to a large portion of the spectrum being under-utilized, and hence a system that increases the efficiency of the spectrum is needed. Spectrum sensing and access are major elements of cognitive radio system design. The main aim of the system is to maximize the utilization of free channels while limiting the amount of collisions with the primary users. These conflicting goals are a challenge to researchers in the field of cognitive radio dynamic spectrum access.

1.3. Thesis Contributions

The main contributions of the thesis can be summarized as follows. A channel access algorithm is proposed that maximizes the utilization of the PU channels compared to other existing access algorithms, such as the one proposed in [1–3]. Moreover, several spectrum sensing techniques that have the potential of increasing the utilization of the PU channels and/or reducing the collision with the PUs are studied extensively. In particular, HMM-decoding, polynomial classifiers and NARX models are used for sensing a PU channel modeled by an HMM.

To the author's best knowledge, the use of classifiers other than HMMs for HMM-generated data in the context of cognitive radio has not been investigated in the literature. The results in fact show a significant improvement in the sensing accuracy with polynomial classifiers, compared to the conventional HMM-decoding approach.

1.4. Thesis Outline

This thesis is organized as follows: Chapter 2 provides a background on cognitive radio and its main research fields. Moreover, it presents some of the primary user activity models adopted in the literature. Furthermore, the necessary background on the

machine-learning techniques used in this thesis to decide the availability of the channels is provided in this chapter.

In Chapter 3, a detailed literature review on spectrum sensing and access is presented. Chapter 4 discusses the proposed access scheme and is followed by the simulation results presented in Chapter 5. Finally, Chapter 6 concludes the thesis, summarizes the main findings and suggests recommendations for further research.

Chapter 2: Background

The electromagnetic radio spectrum is a natural resource used primarily for wireless communications. The spectrum is divided into frequency bands, each allocated to a specific user or application. This fixed allocation policy results in some bands being underutilized, leading to a phenomenon known as *spectrum holes*. A spectrum hole is defined as a frequency band that is not used by its primary user (PU), or the license holder, at a specific time and geographical location [4]. The significance of spectrum holes lies in the scarcity of frequency bands as the demand on wireless communications is in continuous increase. Moreover, it is estimated that the available electromagnetic spectrum is not used 80% of the time [5].

As a consequence, cognitive radio was introduced as a means of accessing the electromagnetic spectrum in order to use spectrum holes efficiently. Cognitive radio (CR) technology was first introduced by Joseph Mitola in 1999 [6, 7]. As defined in [4], a cognitive radio system is a system that is *aware* of its environment and is capable of *learning* and *adapting* to the changes in the environment.

A cognitive radio system consists of two networks: the *primary* and the *secondary* networks. Primary users are license holders and own the full right of using the spectrum band at all times. Dedicated base stations already exist for the primary users, e.g. TV broadcasters. Secondary users on the other hand, share the spectrum band in an opportunistic manner with primary users, since they do not have a license to the band.

The idea lies in the ability of a *cognitive* system to sense the availability of spectrum holes and use the holes until a primary user appears. When a PU appears, the system should be able to reallocate another free band to the SU so that the service is not interrupted. The principle of cognitive radio is illustrated in Figure 1.

Cognitive radio networks (CRNs) could be either *centralized* or *distributed*. In a centralized network, the activities of secondary users are coordinated by base stations. The base station receives spectrum sensing results from the SUs and allocates frequency bands accordingly. Once a primary user is present, it is the duty of the base station to ensure that the SU is reallocated another frequency band such that no interference is caused to the PU, and simultaneously, the SU should not experience any interruption in

the service. In this case, the decisions related to spectrum allocations and reallocations are made solely by the base station.

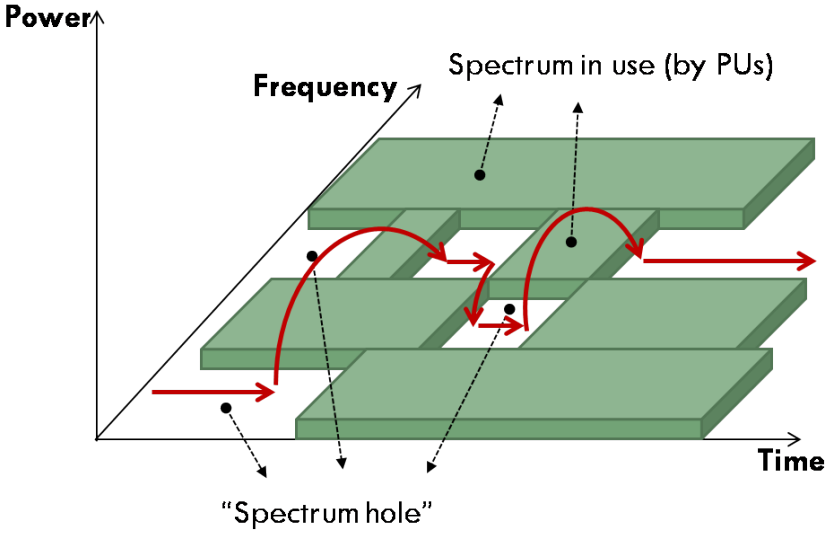


Figure 1: Illustration of the principle of cognitive radio

The main disadvantage of a centralized network is the need for additional infrastructure in conjunction with the existing PU's base stations. Moreover, having a single unit that controls the entire network is considered to be a single point of failure. However, since the decision is made by a single unit, coordination between SUs is considerably simpler. On the other hand, in distributed CR networks, SUs form an ad hoc network without the need for a dedicated base station. In this case, the allocation and reallocation decisions are made by the SUs themselves, which requires significantly more complicated sharing mechanisms. Figure 2 shows the two types of CR network architectures.

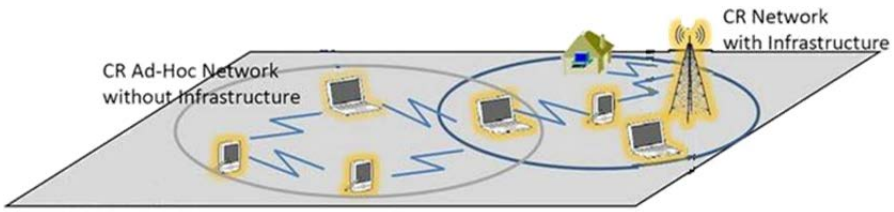


Figure 2: Cognitive radio network architectures

Whether a CR network is centralized or distributed, the main principle employed is *dynamic spectrum access (DSA)*, where PU channels are utilized opportunistically. To design a CR system, two main requirements need to be met: (1) the dynamic access should not have an adverse effect on the usage of the PU's channel (2) the SUs should try to maximize their channel usage to improve spectrum efficiency. In order to satisfy these requirements, several elements need to be taken into consideration, including the decision on the availability of the PU channels, which is the main focus of the thesis.

The rest of the chapter gives an overview of cognitive radio system (CRS) design. In Section 2.1, an introduction to the main functionalities of a CRS is given. Section 2.2 outlines some of the PU activity models adopted in the literature and Section 2.3 discusses a number of artificial intelligence techniques used to decide the availability of the PU channels.

2.1. Functionalities of a Cognitive Radio System

The main functionalities of a cognitive radio system can be categorized into *spectrum sensing*, *spectrum management*, *spectrum mobility* and *spectrum sharing* [8]. Figure 3 summarizes the functionalities of a CRS.

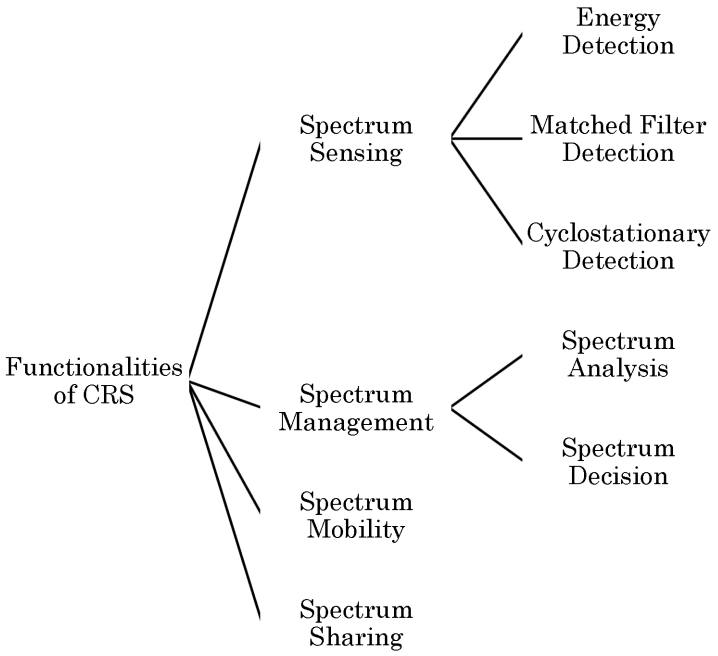


Figure 3: Functionalities of a cognitive radio system

2.1.1. Spectrum sensing. In order to access a free channel, the first step is to determine the available channels. The accuracy with which sensing is done is vital because the primary user should not, under any circumstances, experience any degradation in the performance due to the opportunistic access by the secondary users. There are numerous methods used for spectrum sensing, some of which are [9]:

- *Energy detection*

  As the name suggests, the energy level of the signal is sensed and a decision is made as to whether the primary user is active or not by comparing the output of the energy detector to the noise floor [9]. The main advantage of energy detection is the low implementation cost and reduced computational complexity. However, energy detection is not suitable for spread spectrum systems, where the signal level is low enough to be considered as noise. Moreover, the performance of energy detectors under low signal-to-noise ratio (SNR) is poor, and the distinction between PU and SU transmissions is not possible.

- *Matched filter detection*

  The optimal detector under additive white Gaussian noise (AWGN) is the matched filter [10]. However, this requires the knowledge of the primary user signal statistics, which are not always available.

- *Cyclostationary detection*

  A cyclostationary signal is one whose mean and autocorrelation are periodic. The detector exploits the cyclostationarity property of primary user signals. The cyclostationary detector performs better than the energy detector but requires longer processing time [8].

2.1.2. Spectrum management. After sensing the channels and deciding on their availability, the next step is to decide which channel provides the best quality of service (QoS). Two aspects of spectrum management are *spectrum analysis* and *spectrum decision* [8].

- *Spectrum analysis*

  The quality of the available channels is governed by a number of factors, including the SNR, the holding time, or the time that the SU gets to use the channel

without interruption, the correlation of spectrum holes and the amount of interference and path loss [8, 11]. Analysis of the channels is important in deciding the proper access mechanism.

- *Spectrum decision*

  Given the user requirements and the characteristics of the channel obtained by spectrum analysis, a decision has to be made regarding the optimal choice of a channel. To access a channel, a decision model is required, the most common being stochastic optimization methods, such as Markov decision processes [11, 12].

  Another aspect that has to be considered in spectrum decision is that an SU can use more than one channel simultaneously. The decision in this case has to be made over multiple channels [8].

2.1.3. Spectrum mobility. A secondary user can access a channel only if it is not in use by its primary user.[1] If a channel is currently in use by an SU, and a PU needs to access a channel, the SU needs to vacate the channel immediately such that no interference is caused to the PU. On the other hand, the service to the SU should not be interrupted. This can be achieved by switching the SU to another channel that satisfies the user's requirements, referred to as *spectrum handoff*. There are plenty of elements to be considered in spectrum mobility some of which are the duration of the handoff, and the service degradation introduced by the handoff [8].

2.1.4. Spectrum sharing. So far, the discussion has been limited to a single secondary user. When multiple users are involved, a proper sharing mechanism that controls the usage of the spectrum bands by the secondary users is required.

As discussed earlier, sharing could be achieved in a centralized or a distributed manner. In a centralized network, spectrum sharing is controlled by the CR base station. On the other hand, in a distributed network, the decision is made by each of the users individually.

In a distributed network, sharing could be achieved in a cooperative or non-cooperative manner [8]. In a cooperative sharing scheme, channel access decision is

[1] This is true for overlay spectrum sharing paradigms discussed in Section 2.1.4

made collaboratively. Results of spectrum sensing are shared among users and a decision is made in a way that maximizes the channel usage of the network as whole. A non-cooperative scheme, on the other hand, makes a decision based on each SU individually. The SU would try to maximize its own utilization of the spectrum regardless of other users in a selfish manner. This can reduce the spectrum utilization of the network as a whole, but it also minimizes the information shared between users.

Another spectrum sharing principle is based on the access technique used. An SU can access a channel in three ways: *overlay*, *underlay* or *interweave* spectrum access [13,14]. In underlay spectrum sharing, the SU transmissions are concurrent with the PU transmissions, using spread spectrum techniques that utilize high bandwidths and are considerably more complex, such that the SU transmissions are perceived as noise by the primary user. In overlay spectrum sharing, SUs use prior knowledge of the primary user transmissions to help improve the PU communication and simultaneously for SU transmissions. In interweave spectrum sharing, the SU is allowed to use the frequency bands only if the primary user is not active. Hence, a good spectrum access technique is needed to avoid interference to the primary user. Throughout this thesis, interweave spectrum access will be adopted.

2.2. Primary User Activity Models

A good cognitive radio system design needs to have an accurate PU activity model to ensure that PUs are not affected by the SU transmissions. It is worth noting that no single model can describe all patterns of primary user activity. This is because different wireless networks have different patterns of PU activity.

The most commonly used primary user activity models fall under the umbrella of either Markov processes or queuing theory [15]. The following subsections provide an overview of these models.

2.2.1. Markov processes. A Markov process is one whose future is independent of the past, i.e., given that the system is in a particular state, the next state of the system will depend only on the current state, but not on previous states [16].

Several PU activity models are based on Markov processes. An example of such models is a two-state Markov chain, where a PU channel can be in one of two states: *busy* or *idle*. The Markov chain could be either discrete or continuous. A discrete-time Markov chain (DTMC) is characterized by transition probabilities p_{ij}, where i and j represent the current and next states respectively. On the other hand, in a continuous-time Markov chain (CTMC), the time it takes to transition from one state to another is exponentially distributed, with rate λ for the transition from *idle* to *busy* and rate μ for the transition from *busy* to *idle*.

Another model based on Markov processes is the three-state Markov chain proposed in [17], where the PU channel could be in one of three states: *busy*, *idle* or *occupied by SU*. The advantage of this model over the two-state model is that the system takes into account the interference caused by other SUs. This, however, does not provide a model for the interference among SUs.

Unlike the Markov chains discussed so far, in hidden Markov models (HMMs) the states of the channel are hidden. Each hidden state emits a set of observable symbols, from which the state of the channel needs to be deduced. HMMs can be further classified based on their emissions, which can be discrete or continuous. In [18], the PU activity is modeled by an HMM with discrete observations, while modeled by an HMM with continuous observations in [1]. Although HMMs are a powerful tool, modeling a PU activity by HMMs is limited in the literature.

2.2.2. Queuing theory. The activity of the PU can be modeled by a queuing system. Before going through the details of the queuing systems used in the literature, it is useful to introduce the notation used in queuing theory. A queuing system is typically denoted by $a/b/m/K$, where a is the type of the arrival process, b is the distribution of the service time, m is the number of servers, and K is the maximum number of users [16].

A commonly used queuing model for the PU's activity is the M/G/1 model, where M represents a Poisson arrival process with independent and identically distributed (iid) exponential interarrival times, and G represents iid service times of a

general distribution[2]. The packet arrival rate is denoted by λ, and the service rate is given by $1/E[D]$ where $E[D]$ is the expected value (or the average) of D, and D is the medium access delay, or the time it takes a packet to gain access to the channel.

Although the queuing model is simple to use and is widely used in the literature, it does not capture the small variations in the PU's activity. Furthermore, the model does not take into account any correlation in the PU's activity [15].

2.3. Artificial Intelligence Techniques

To find or predict the availability of a primary user, different spectrum-sensing techniques can be used. Artificial intelligence (AI) is a broad class of spectrum-sensing techniques, where *machine learning* algorithms are used to learn the patterns of the PU activity. This section gives a brief background on some of these techniques, namely hidden Markov models (HMMs), polynomial classifiers and nonlinear autoregressive with exogenous inputs (NARX) models.

2.3.1. Hidden Markov models and their parameters.

Unlike several machine learning techniques, the notion of time is inherent in hidden Markov models, i.e., states at a particular time instant are dependent on the previous state [19].

Before going into the details of HMMs, it is essential to describe the Markov property. The Markov property states that a random process at a particular state at any time instant t is only dependent on the previous state (the state at $t-1$) regardless of any other states the process went through to reach the state at $t-1$ [16]:

$$\begin{aligned} P[X(t) &= x_t | X(t-1) = x_{t-1}, \ldots, X(1) = x_1] \\ &= P[X(t) = x_t | X(t-1) = x_{t-1}] \end{aligned} \tag{1}$$

where $P[.]$ represents the probability of an event, X is a random process and x is a realization of that process.

If the random process is discrete-valued, the Markov process is called a Markov *chain*. Let S represent the states of the Markov chain, q_t represent the state at time t and

[2]When K is not specified, the number of users is unlimited

N denote the total number of states, then $\mathbf{A} = \{a_{ij}\}$, the transition probability matrix, is given by

$$\{a_{ij}\} = P[q_{t+1} = S_j | q_t = S_i], \qquad 1 \leq i, j \leq N. \tag{2}$$

Figure 4 shows the state diagram of a 3-state Markov chain, with marked transition probabilities.

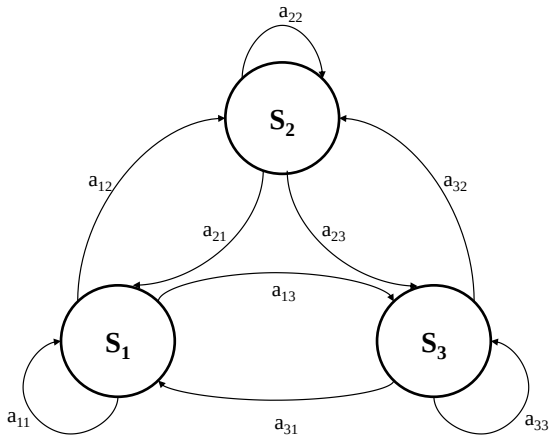


Figure 4: An example of a 3-state Markov Chain

In a hidden Markov model, each state emits one or more observed states, which are a probabilistic function of the *hidden* states [20]. The observation symbol probability matrix, denoted by $\mathbf{B} = \{b_j(k)\}$, is given by

$$\begin{aligned} b_j(k) = P[V_k \text{ at } t | q_t = S_j], \quad & 1 \leq j \leq N \\ & 1 \leq k \leq M \end{aligned} \tag{3}$$

where V represents the observations, and M is the total number of observations for the discrete case. Furthermore, the initial state distribution, denoted by $\boldsymbol{\pi} = \{\pi_i\}$, is given by

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N. \tag{4}$$

A typical HMM is therefore characterized by a set of parameters $\theta = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$.

Eq. (3) is true when the observed sequence is discrete with M distinct observations. However, if the observed symbols are continuous, and if in addition the symbols

follow a Gaussian distribution, $b_j(k)$ is redefined as

$$b_j(O_t) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(O_t-\mu_j)^2}{2\sigma_j^2}\right), \qquad 1 \leq j \leq N \tag{5}$$

where O_t is the observed symbol at time t, μ_j and σ_j are the mean and standard deviation of the observations given state j respectively. The HMM can now be modeled by $\theta = (\mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$, where $\mathbf{B}$, the observation probability matrix has been replaced by $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_N]$ the mean vector and $\boldsymbol{\Sigma} = [\sigma_1^2, \sigma_2^2, \ldots, \sigma_N^2]$ the variance vector since M, the total number of observations, is now infinite.

For the above formulation to be valid, two conditions must be satisfied [21]. First, the current state should be independent of all other states given the previous state. Second, the observation given a state is independent of all other observations of the state.

2.3.2. Linear discriminant classification. The main principle in linear discriminant classification is that the samples of different classes are linearly separable. While this might not be true in several cases, data can be projected to higher dimensions such that the decision surface is linear, as illustrated in subsequent sections. One of the advantages of a linear classifier is that no assumption is made on the distribution of the data [19].

A linear discriminant function of a d-dimensional feature vector $\mathbf{x}$ takes the form:

$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 \tag{6}$$

where $\mathbf{w}$ is the weight vector, w_0 is the bias, and $(.)^t$ is the transpose operator. In general, if there are N classes, N discriminant functions are needed, one for each class. For a two-class problem, the decision can be made as follows: decide class S_1 if $g(\mathbf{x}) > 0$ and class S_2 if $g(\mathbf{x}) < 0$.

To simplify the notation, Eq. (6) can be re-written as [19]:

$$g(\mathbf{x}) = \mathbf{a}^t\mathbf{y} \tag{7}$$

where **a** is the augmented weight vector given by

$$\mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} \tag{8}$$

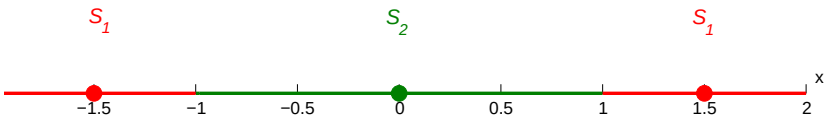
and **y**, the augmented feature vector, is given by

$$\mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}. \tag{9}$$

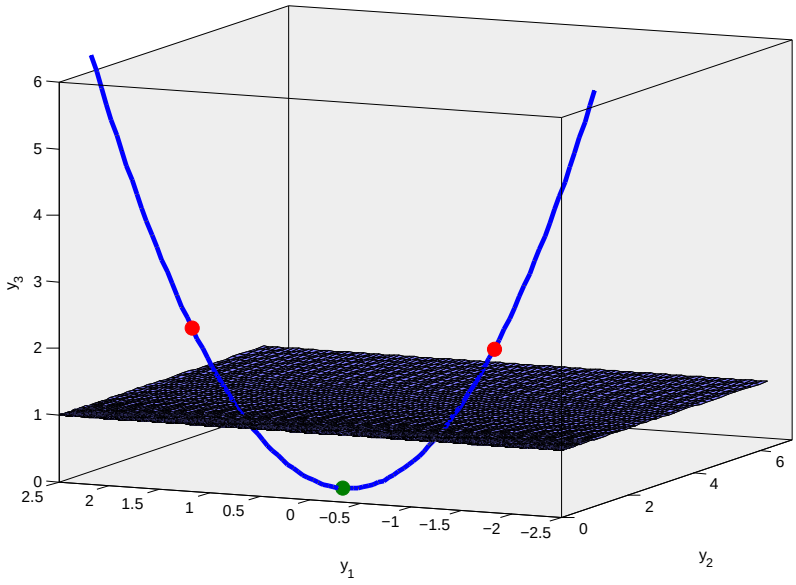
2.3.2.1. Feature expansion. As discussed earlier, assuming that samples of different classes are linearly separable is not always true. To use linear discriminant functions, data samples that are non-linearly separable are projected to higher dimensions, where they can be separated by a linear hyperplane. Figure 5 shows the mapping of a one-dimensional feature vector to three dimensions. It can be seen clearly in the figure that although the data from the two classes are non-linearly separable, projecting the data onto a higher dimension allows the data to be separated by a linear plane.

One class of mapping functions that project the data onto higher dimensions are *polynomial* discriminant functions. Let $\hat{d}$ denote the dimension of the feature vector after mapping. Suppose we have a 1-dimensional feature vector ($d = 1$), a quadratic discriminant function ($\hat{d} = 3$) is given in the form:

$$g(x) = a_0 + a_1 x + a_2 x^2 \tag{10}$$



(a) 1-D feature vector



(b) 3-D mapping

Figure 5: Mapping of 1-D feature vector to 3-D

where, equations (8) & (9) are replaced by:

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \tag{11}$$

$$\mathbf{y} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}. \tag{12}$$

In general, a polynomial discriminant function of order n ($\hat{d} = n+1$) is given in the form

$$g(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n. \tag{13}$$

For feature vectors of higher dimensionality ($d > 1$), cross terms are also taken into account. For example, a quadratic discriminant function for a d-dimensional feature vector is given by

$$g(x) = a_0 + \sum_{i=1}^{d} a_i x_i + \sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} x_i x_j. \tag{14}$$

It should be noted that the number of terms involved in the computation of discriminant functions with feature vectors of high dimensionality increases exponentially with the order of the polynomial [19]. A d-dimensional feature vector would require d^n terms, which makes it computationally unfeasible for high values of d and n.

2.3.3. Nonlinear autoregressive with exogenous inputs (NARX) model. The use of artificial neural networks (ANNs) as a machine learning technique has several desirable characteristics, including the ability to map any input to any output. *Neural networks* are inspired by the way the brain works, where many nerve cells, or *neurons*, are activated in a specific manner to perform a particular function. A typical neural network consists of an *input* layer, one or more *hidden* layers, which consist of neurons with non-linear activation functions, and an *output* layer.

One type of neural networks that represent dynamic networks, or networks that change with time, is the *nonlinear autoregressive with exogenous inputs (NARX)* model. Unlike static neural networks, the output of a NARX network is fed back to the input [22]. Figure 6 shows a NARX network with 2 input delays, 2 output delays, one hidden layer with 3 neurons and an output layer.

A NARX network can be mathematically expressed as [22]:

$$y(n+1) = f\left(y(n), \ldots, y(n-q+1), u(n), \ldots, u(n-q+1)\right) \tag{15}$$

where $y(n+1)$ is the output at time index $n+1$, $u(n)$ is the input, and q is the number of delays in the system. Eq. (15) can be expressed in terms of the synaptic weights, or the strength of a connection between neurons, as follows [23]:

$$y(n+1) = f\left(\sum_{j=1}^{n_H} w_{kj}(n)\left(\sum_{i=1}^{q} w_{u,ji} u(n-i+1) + \sum_{i=1}^{q} w_{y,ji} y(n-i+1) + w_{j0}\right)\right) \tag{16}$$

where w_{ji} and w_{kj} are the weights that connect the hidden and input layers, and the output and hidden layers respectively, w_{j0} is the weight of the bias, n_H is the number of neurons in the hidden layer, and the subscripts u and y are used to distinguish the input and output weights.

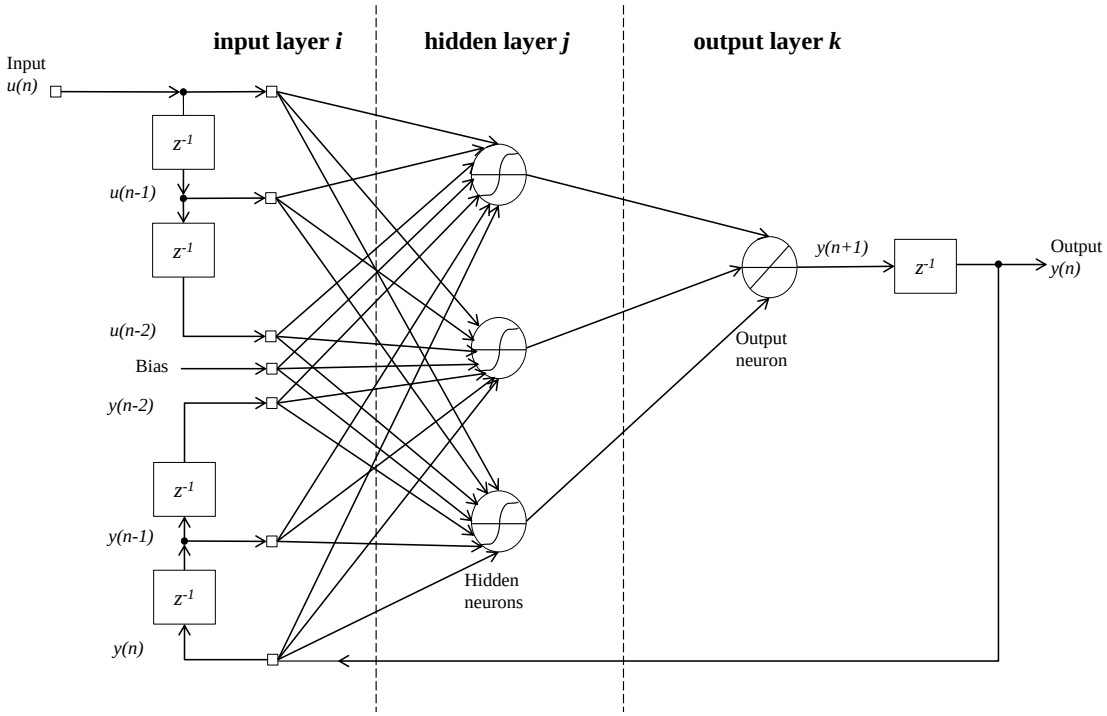


Figure 6: An example of a NARX network

Chapter 3: Literature Review

Spectrum management is one of the key elements in cognitive radio system design. As explained in Section 2.1.2, spectrum management is required to analyze the available channels and decide on the channel that best suits the user's requirements. The problem of spectrum management has been addressed in the literature in one of the three approaches: *optimization*, *game theory*, and *machine learning* [11]. This chapter provides a detailed literature review of the three different approaches. Additionally, spectrum sensing using energy detection is briefly discussed to get an insight of the most common spectrum sensing technique.

3.1. Optimization

The problem of spectrum management can be formulated as an optimization problem, where an objective function is to be maximized or minimized under certain constraints. Optimization techniques can be classified into [11]:

- *Closed-form solution*

  The Lagrangian method can be used to solve constrained optimization problems.

- *Mathematical programming*

  Techniques including linear, non-linear, convex, dynamic and stochastic programming can be used to model the problem of spectrum management.

- *Integer/combinatorial optimization*

  When the target is to optimize with respect to a parameter that has only integers, such as the modulation index, integer programming can be used. Similarly, parameters such as channel allocation have a combinatorial nature. Such problems are solved using relaxation and decomposition, enumeration, cutting planes, and solutions of the knapsack problem.

Some of the optimization techniques for spectrum management are discussed in [24–31]. In [24], resource allocation is done based on swarming mechanisms. Given a network of secondary users, the goal is to minimize an objective function with respect to a vector of frequency bands that each SU occupies. The minimization is also constrained by the QoS requirements of the SUs expressed in terms of the signal to noise

and interference ratio (SNIR). The probability that the SNIR is less than a threshold η should be less than a certain probability P_ε.

Let x_k be the frequency band chosen by SU k and let the total number of SUs in the network be C. The optimization problem can be formulated as [24]:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & J(\boldsymbol{x}) \\ s.t. \quad & \text{Prob}\{\widehat{\text{SNIR}}(x_k) < \eta\} \leq P_\varepsilon, \quad k = 1, \ldots, C \end{aligned} \tag{17}$$

where $J(\boldsymbol{x})$ is the objective function given by [24]:

$$J(\boldsymbol{x}) = \sum_{k=1}^{C} I_k(x_k) + \frac{1}{2} \sum_{k=1}^{C} \sum_{l=1}^{C} a_{kl} J_{ar}\left(|x_l - x_k|\right) \tag{18}$$

where the first term represents the sum of the interference perceived by the SUs and the second term is an attraction/repulsion term that represents the swarming mechanism of the frequency allocation.[1]

In [25], spectrum sharing is formulated in terms of the interference temperature, or the radio frequency power per unit bandwidth measured at a receiving antenna. To relax the assumption of a convex objective function, particle swarm optimization (PSO) is used. Moreover, simulated annealing (SA) is incorporated in the proposed algorithm to prevent the swarm particles from getting trapped into a local minimum.

In [29], three variants of particle swarm optimization are compared, namely the binary PSO, the socio-cognitive PSO and derivation 0, with the objective to maximize the throughput of secondary links, under the interference constraint.

Swarm intelligence portrayed in [24, 25] is one of several biologically-inspired solutions to problems in communications. In the context of spectrum management, several algorithms inspired by insect colonies have been developed [26]. In [27], a biologically-inspired spectrum sharing (BIOSS) algorithm is proposed, where decentralized spectrum sharing is performed without inter-user coordination. However, a high probability of collision between the SUs decreases spectrum utilization. This problem is addressed in [28], where an enhanced BIOSS (e-BIOSS) algorithm is proposed. The e-BIOSS algorithm enables SUs to select the appropriate channels and simultaneously

[1] For more details on Eq. 18, please refer to [24].

avoid collision between SUs. The SUs are gravitated towards channels that have minimum power. Moreover, a binary learning factor is introduced, which decides whether to access a certain channel or not.

In [32], improved throughput and spectrum sensing capabilities are achieved by formulating a Lagrange dual optimization problem. The proposed access algorithm has the optimal power allocation strategy and target detection probability. A Lagrange function is also introduced in [31], with the aim of reducing the computational complexity of existing sensing-based access schemes.

3.2. Game Theory

Game-theoretic approaches are commonly used in resource allocation problems. The general principle in game theory lies in the competition between players (or users) each having their own *strategy*, trying to maximize their *utilities*. There are different types of games including *non-cooperative*, *cooperative* and *stochastic* games [11].

- *Non-cooperative games*

  In the case where information is restricted to local information, there is no option but to adopt a *non-cooperative* game, however, the overall outcome is less efficient. Some concepts of *non-cooperative* games include the Nash equilibrium and Pareto optimality [11].

- *Cooperative games*

  In *cooperative games*, players do not actually cooperate with each other. However, the cooperation is enforced by a third party (e.g., a regulation authority) in the form of penalties, etc. Some examples of *cooperative* games are the bargaining game and the coalition game.

- *Stochastic games*

  Stochastic games are used when the information is not deterministic but rather statistical. One important example is the Markov decision process (MDP), where the current state of the game depends on the previous state and the current actions of the players.

In [33], a double-auction model from the microeconomic theory is used in modeling the trading of TV bands among TV broadcasters and wireless regional area network (WRAN) service providers. For WRAN service providers, the problem of spectrum bidding and pricing is formulated as a non-cooperative game. The Nash equilibrium is obtained as the solution.

Another spectrum access scheme based on auction models has been developed in [34]. The objective is to maximize the revenue of the service provider, as well as the satisfaction of the SUs under the condition of imperfect channel sensing. In order to do so, a sealed-bid first-price auction is proposed. Results show a better performance compared to other spectrum trading methods.

In [35], a distributed, collaborative spectrum sharing approach is proposed with minimum coordination overhead. This is achieved by the implicit coordination between SUs through a set of predefined rules. Five rules are proposed where a trade-off is made between performance, and implementation and computational complexity. Results show superior performance compared to other existing schemes.

In [36], both cooperative and non-cooperative spectrum access schemes are considered. They show that the optimal scheme for both cases is solely based on threshold policies. Moreover, for the non-cooperative case, access strategies are developed so that the system reaches a Nash equilibrium.

In [37], a spectrum sensing and access scheme is proposed where the channel access scheme is modeled as a multi-armed bandit problem. Different scenarios are studied including the case of partial channel sensing, where the sensing is done for only a fraction of time. Moreover, detailed derivations are provided for the computational complexity for the cases of full and partial channel sensing.

The following sections will focus on the spectrum sensing techniques investigated in the literature.

3.3. Energy Detection

Energy detection is one of the most commonly adopted spectrum sensing techniques in the literature due to its simplicity and mathematical tractability. Two hypothe-

ses H_0 (the channel is available) and H_1 (the channel is busy) are made. The received signal is modeled as [38]:

$$x[n] = \begin{cases} v[n] & \text{if } \mathrm{H}_0 \text{ holds} \\ v[n] + hs[n] & \text{if } \mathrm{H}_1 \text{ holds} \end{cases} \tag{19}$$

where $v[n]$ is additive white Gaussian noise (AWGN), with zero mean and standard deviation σ_v, $s[n]$ is the PU signal, and h is the channel gain.

The energy of the signal $x[n]$, denoted by $y[k]$ is given by [38]:

$$y[k] = \sum_{n=0}^{N-1} |x[n+kN]|^2 \tag{20}$$

where k is the time index of the energy estimator and N is the total number of received samples. The decision on the availability of the channel is made based on a threshold γ, above which the channel is assumed to be busy.

Numerous variants of the basic form of energy detection were investigated in the literature, some of which are discussed in this section.

In [38], a novel algorithm is proposed, which introduces an objective function that depends on a test statistic that contains information about the availability of the channel. The objective function performs weighted cooperative sensing, and also spectrum sensing locally at each SU. Results show that the proposed algorithm outperforms other state-of-the-art energy detection algorithms in terms of probability of miss-detection.

In [39, 40], energy detection spectrum sensing for a cooperative scenario is investigated under fading conditions. A novel algorithm is proposed where an extra sensing cycle is performed locally at the SUs to overcome the problem of outdated sensing results caused by the delay introduced due to cooperative sensing. The proposed scheme compromises between sensing accuracy and computational complexity.

3.4. Machine Learning

Machine learning techniques are used in spectrum sensing to decide the availability of primary users. The advantage of machine learning over traditional cooperative sensing schemes is that it is more adaptive, since it does not require prior knowledge of the environment [41]. Moreover, machine learning techniques are also used in spectrum management to predict the behavior of the primary user based on its past activities. The three main categories of machine learning are [11]:

- *Supervised learning*

  In supervised learning, the classifier is first trained using a set of training data, where the output to a given input is known. Once the classifier is trained, it can determine the output corresponding to an input of a new set of data. Methods including support vector machines (SVMs) and weighted k-nearest neighbors (kNNs) are used to determine the availability of primary user channels [41].

- *Unsupervised learning*

  Unsupervised learning uses clustering techniques to identify the class of a given input, without the training stage. Some of the methods used are the k-means and Gaussian mixture models (GMMs) [41].

- *Reinforcement learning*

  Similar to unsupervised learning, reinforcement learning does not require a training stage. However, in reinforcement learning, the agent explores all possible actions and rewards in a trial-and-error manner. This optimizes the long-term on-line performance of the system [11].

3.4.1. Hidden Markov models. Hidden Markov models are one of the important machine learning techniques that are widely used for prediction. HMMs have been used for prediction in various fields, including stock markets and speech recognition because of their solid theoretical foundation and tractability [20]. In cognitive radio spectrum management, in order to gain access to the spectrum, the primary user channel usage pattern has to be learned first, then, based on the learned pattern, predictions about future PU patterns can be made.

The use of HMMs for primary user pattern prediction is discussed in [1–3, 12,

42–45]. In [12], the SU first learns the pattern of the primary user using a hidden Markov model. The PU is modeled as a continuous-time Markov chain (CTMC) whose parameters are obtained using a gradient method. The SU then gains access to the channel using a partially-observable Markov decision process (POMDP), which is a form of stochastic games. Simulation results show that the proposed scheme provides efficient access to spectrum holes, and the interference to the primary users is below the target limit. The proposed scheme also outperforms a heuristic algorithm without any learning capability.

Similar to [12], dynamic spectrum access in [46] is formulated as a Markov decision process and is solved using a linear programming relaxation and primal-dual index heuristic algorithm. The proposed scheme minimizes the energy consumption and decreases the frame error rate and thereby improving the QoS.

In [42], the cognitive radio system is modeled as an HMM, and the channel parameters are estimated using an expectation maximization algorithm. Numerical results show that the proposed algorithm successfully estimates the true channel parameters.

In [43], the Baum-Welch algorithm is used to determine the HMM parameters and the results are tested on the 450-470 MHz band in Australia. The proposed model determines the optimal number of SUs to operate in the spectrum band by examining the channel usage that results in maximum reward.

In [1], the HMM predictions of the primary user channel availability are tested with real spectrum data. Results show that HMM-based spectrum sensing outperforms energy detection, especially in cases where the SNR is low. The use of HMMs for spectrum sensing is also discussed in [47].

3.4.2. Polynomial classifiers. Polynomial classifiers have been used in various machine learning applications including communications. Classification of modulation schemes, for example, is one of the applications that use polynomial classifiers [48].

The use of polynomial classifiers for spectrum sensing is discussed in [49–52]. In [49], linear and polynomial classifiers are compared, and their performance is evaluated based on the probability of detection of a PU. Results show that linear and poly-

nomial classifiers are comparable, with high detection rates even at low signal-to-noise ratios (SNRs). A variant of these classifiers is proposed in [50,51], where energy, coherent or cyclostationary features are used. It is shown in [51] that cyclostationary-based schemes are the most reliable but require longer sensing periods. Moreover, in [50], the system is tested under fading conditions.

In [52], a hybrid cooperative sensing technique that combines energy detection and cyclostationary detection is proposed, and the decision is made at a fusion center. Results show that using the hybrid scheme outperforms energy detection in terms of accuracy and cyclostationary detection in terms of computational complexity.

In this report, polynomial classifiers are used to learn the availability of a PU modeled by an HMM, a case that has not been studied before in the literature. In particular, HMM-generated data is usually decoded by the conventional Viterbi algorithm, which results in high collision probabilities, as discussed in Chapter 5. The use of polynomial classifiers is expected to enhance the performance of the system. Furthermore, a channel access algorithm is proposed, which improves the spectrum efficiency compared to the one proposed in [1].

Chapter 4: Proposed Channel Access Scheme

In a cognitive radio system, a secondary user first performs spectrum sensing and decides whether a PU channel is available or not. If more than one SU is present, the decision is made in a centralized manner, where a fusion center makes a global decision on the availability of the channel. Once a decision is made, an SU can either use the channel or not, based on the sensing results and on the access rules of the system.

A generalized block diagram of the proposed system is illustrated in Figure 7.

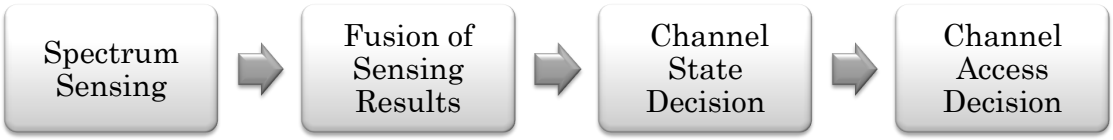


Figure 7: Block diagram of the proposed system

The function of each block is as follows:

- *Spectrum sensing*

  Secondary user(s) sense the PU channel(s) to be used for transmission. The energies of the received signals detected by the sensors are then sent for fusion in the case of multiple secondary users, or used directly to decide the state of the channel in the case of a single SU.

- *Fusion of sensing results*

  In the case of several secondary users, it is required that sensing results are combined in such a way that it is used for making a global decision about the availability of the channel(s). In a centralized network, the results are sent to a fusion center, or the base station.

- *Channel state decision*

  Once the results are combined, a global decision about the availability of the channel has to be made. Given the energy level, the system determines whether the channel is in the 'ON' state (where the channel is occupied) or the 'OFF' state (where the channel is available).

- *Channel-access decision*

  If a channel is determined to be available, the SU has two options, either to access the channel or not. The decision depends on several factors, including the amount of anticipated collision with the primary user if the SU was to access the channel, and the efficiency of channel usage if the SU was to abandon the channel.

4.1. Primary User Model and Channel Availability

The system consists of P primary users and one secondary user. Each primary user is assumed to occupy one channel and hence the number of channels corresponds to the number of primary users. The primary user is modeled by a two-state discrete-time Markov chain ($N = 2$) with initial state transition matrix $\mathbf{A}$ and initial probability mass function (pmf) vector $\boldsymbol{\pi}$. The state of the primary user (or the channel) is 0 when the primary user is off (or the channel is available) and 1 when the primary user is on. The observed sequence is a series of energy levels (in dBm) that are Gaussian distributed with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$. For simplicity, errors in the measured energy levels are assumed to be negligible. Each of the P channels is sensed periodically, and a vector of energy levels is obtained. Given the observed sequence of energies, the HMM estimates the most likely state sequence, as well as the HMM parameters $\mathbf{A}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ by the procedure outlined in the following sections.

4.1.1. Parameter estimation. Given an observed sequence $O = \{O_1 O_2 \ldots O_T\}$, where T is the length of the sequence, it is required to estimate the model parameters that most likely produced the sequence. The procedure for estimating the HMM parameters is known as the *Baum-Welch* or the *forward-backward* algorithm.

Considering every possible model that could produce the observed sequence is computationally prohibitive, hence two variables, namely the *forward* and *backward* variables, are introduced. The forward variable, denoted by $\alpha_t(i)$ is defined as [20]:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \theta) \tag{21}$$

and is computed recursively as follows:

$$\alpha_{t+1}(j) = \begin{cases} \pi_j b_j(O_1) & t = 0 \text{ and } 1 \leq j \leq N \\ \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) & 1 \leq t \leq T-1 \text{ and } 1 \leq j \leq N \end{cases} \tag{22}$$

Similarly, the backward variable, denoted by $\beta_t(i)$ is defined as [20]:

$$\beta_t(i) = P(O_1 O_2 \cdots O_t | q_t = S_i, \theta) \tag{23}$$

and is computed as follows:

$$\beta_t(i) = \begin{cases} 1 & t = T \text{ and } 1 \leq i \leq N \\ \sum_{i=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) & t = T-1, T-2, \ldots, 1 \text{ and } 1 \leq i \leq N \end{cases} \tag{24}$$

Let $\xi_t(i,j)$ denote the probability of being in state S_i and S_j at times t and $t+1$ respectively, given the observation sequence and the model [20]:

$$\begin{aligned} \xi_t(i,j) &= P\left(q_t = S_i, q_{t+1} = S_j | O, \theta\right) \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\theta)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{25}$$

Furthermore, let $\gamma_t(i)$ denote the probability of being in state S_i at time t, given the observation symbol and the model, i.e. [20]

$$\begin{aligned} \gamma_t(i) &= P(q_t = S_i | O, \theta) \\ &= \frac{\alpha_t(j) \beta_t(j)}{P(O|\theta)} \\ &= \frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)} \end{aligned} \tag{26}$$

The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be estimated as [20],

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} \gamma_t(j).O_t}{\sum_{t=1}^{T} \gamma_t(j)} \tag{27}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{t=1}^{T} \gamma_t(j).(O_t - \mu_j)^2}{\sum_{t=1}^{T} \gamma_t(j)} \tag{28}$$

The elements of the state transition matrix $\mathbf{A}$ are computed as follows [20]:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{29}$$

Finally, $\boldsymbol{\pi}$ can be estimated as [20],

$$\hat{\pi}_i = \gamma_1(i) \tag{30}$$

The parameter estimation procedure, or the Baum-Welch algorithm, can be summarized as follows. First, π_i, a_{ij}, μ_j and σ_j^2 are initialized using rough estimates. The improved estimates are then calculated using Eqs. (27)-(30) and the process is repeated until convergence is achieved. The Baum-Welch algorithm is considered to be a generalized *expectation-maximization (EM)* algorithm. The computational complexity of the Baum-Welch algorithm is $O(S^2T)$, where $S = 2$ since the system can be in one of 2 states, and T is the total number of slots. Figure 8 shows the Baum-Welch algorithm [19].

The estimated HMM parameters serve as initial model parameters that are constantly updated, as discussed in Section 4.2.

It is worth noting that in general, parameter estimation is done for observations that are a mixture of Gaussians, rather than a single Gaussian distribution for each state. However, since the adopted PU model has a single Gaussian for each state, this prior knowledge is used for parameter estimation to reduce unnecessary computations.

Baum-Welch Algorithm

1: **initialize** π_i, a_{ij}, μ_j, σ_j^2, convergence criterion κ, $z = 0$
2: **repeat**
3: $z = z + 1$
4: compute π_i, a_{ij}, μ_j, σ_j^2, by (27)-(30)
5: $\pi_i(z) \leftarrow \hat{\pi}_i(z)$
6: $a_{ij}(z) \leftarrow \hat{a}_{ij}(z)$
7: $\mu_j(z) \leftarrow \hat{\mu}_j(z)$
8: $\sigma_j^2(z) \leftarrow \hat{\sigma}_j^2(z)$
9: **until** $\max_{i,j} \Big[\pi_i(z) - \pi_i(z-1), a_{ij}(z) - a_{ij}(z-1), \mu_j(z) - \mu_j(z-1), \sigma_j^2(z) - \sigma_j^2(z-1)\Big] < \kappa$

Figure 8: Baum-Welch algorithm

4.1.2. Viterbi algorithm. Given a sequence of observed symbols $O = \{O_1 O_2 \dots O_T\}$, it is required to know the state sequence $Q = \{q_1 q_2 \dots q_T\}$ that is most likely to produce the observed sequence. The objective is therefore to choose the path that maximizes $P(Q|O,\theta)$, which is equivalent to maximizing $P(Q,O|\theta)$. Let $\delta_t(i)$ denote the highest probability of a state and observation sequence up to time t that ends with state S_i [20]:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_t = i, O_1 O_2 \dots O_t | \theta]. \tag{31}$$

Furthermore, let $\psi_t(j)$ be the vector that stores the argument that maximizes (31). The procedure with which the most likely state sequence is found, based on dynamic programming methods, is known as the Viterbi algorithm, and is shown in Figure 9 [20]. Similar to the Baum-Welch algorithm, the computational complexity of the Viterbi algorithm is $O(S^2 T)$.

4.2. Proposed Channel Access Algorithm

Spectrum sensing is performed on every channel and the state of the channel is determined by an HMM as described in Section 4.1. Since the system consists of P primary users, we require P HMMs, one for each channel.

Viterbi Algorithm

1: $\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$
2: $\psi_1(i) = 0$
3: **for** $t = 2$ **to** T **do**
4: **for** $j = 1$ **to** N **do**
5: $\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t)$
6: $\psi_t(j) = \arg\max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right]$
7: **end for**
8: $q_T^* = \arg\max_{1 \leq i \leq N} [\delta_T(i)]$
9: **for** $t = T - 1$ **to** 1 **do**
10: $q_t^* = \psi_{t+1}\left(q_{t+1}^*\right)$
11: **end for**
12: **end for**

Figure 9: Viterbi algorithm

The decision whether or not to access the channel is based on the current state of the channel only. Other access algorithms, for instance, the one proposed in [1], suggest that the decision should be made based on the current and future states of the channel. In Chapter 5, it is shown that transmission based on the current state outperforms the transmission based on current and future states, with no prominent increase in the probability of collision with the primary user.

The rate at which the channel is sensed and a decision is made should be proportional to the latency time, which is the time required by the secondary user to respond to a change in the state of the channel. Specifically, the time the SU takes to leave a channel when a PU is detected must be strictly less than the sensing period, denoted by τ. In order to increase the probability of predicting the state of the channel, the SU does not transmit before a couple of sensing periods. This allows the HMM to learn the pattern of the primary user and to predict the current state of the channel. The frequency with which the sensing is performed depends on the architecture of the SU transceiver as well as the maximum time of interference permitted by the primary user. For simplicity, it is assumed that the maximum allowed interference time is longer than the sensing period τ.

In [12], the authors propose a frame structure, shown in Figure 10, which will be adopted in this thesis.

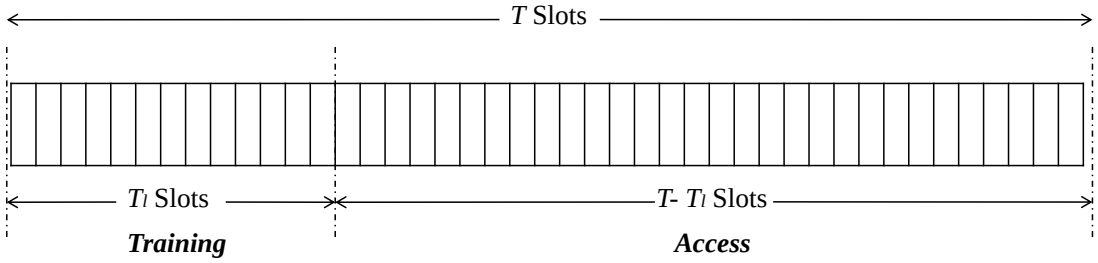


Figure 10: Frame structure

The time frame is divided into timeslots of length τ, and the total number of slots in a frame is T. Let t denote the time at the beginning of a time frame. For the first $t + m\tau$ timeslots, where $m = 1, 2, \ldots T_l$ and T_l is the number of slots used for learning, the channel is sensed and the energy levels are stored in a vector ρ. At the time instant $t + T_l$, the HMM is trained and the model of the primary user is estimated. This process is done for each of the P channels, and the channel whose parameters correspond to the least active primary user (u) is chosen for the transmission. For the next $T - T_l$ timeslots, the state of the channel is estimated. The SU will not transmit unless the current timeslot is free, i.e., current state is 0. Figure 11 summarizes the proposed access algorithm.

Consider an example of a cognitive radio system with 2 PUs. The HMM parameters are given by:

$$\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \tag{32}$$

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} \tag{33}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_0 & \mu_1 \end{bmatrix} \tag{34}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_0^2 & \sigma_1^2 \end{bmatrix} \tag{35}$$

Figure 12 shows a realization of 2 PU channels with hidden states marked for each timeslot, and the corresponding transition matrix $\mathbf{A}$.

Channel Access Algorithm

1: **for** $m = 1$ **to** T_l **do**
2: **for** $i = 1$ **to** P **do**
3: sense channel i and record its energy level
4: **end for**
5: **end for**
6: estimate the HMM parameters of each channel using the Baum-Welch algorithm (Figure 8)
7: choose channel u corresponding to the least active PU
8: **for** $m = T_l + 1$ **to** T **do**
9: estimate the state of the channel u using the Viterbi algorithm (Figure 9)
10: **if** state $= 0$ **then**
11: transmit in slot m
12: **end if**
13: **end for**

Figure 11: Proposed channel access algorithm

The two elements that are of particular interest are a_{00} and a_{11}. The larger the value of the element a_{00}, the more likely the channel will stay in the 'OFF' state, and the lower value of a_{11}, the more likely the PU is to vacate the channel. As seen in the figure, channel 1 corresponds to the least active PU, and hence channel 1 is used for transmission. For the rest of the timeslots, the Viterbi algorithm estimates the state of the channel, and if the channel is available, the SU is allowed to start transmission.

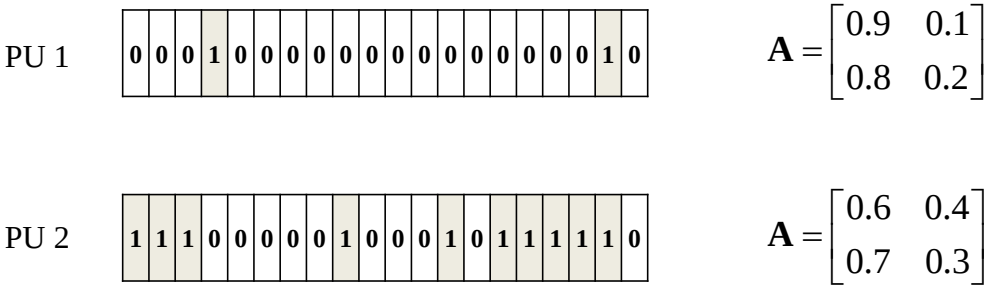


Figure 12: Channel usage pattern of 2 primary users

4.2.1. Supervised learning techniques for channel-state decision. An important element of the channel access algorithm is deciding whether the channel is avail-

able or not. The Viterbi algorithm that is commonly used for HMM-decoding results in high collision probabilities with the PU, as demonstrated in Chapter 5. Therefore, other machine learning techniques need to be investigated in order to achieve lower collision probabilities.

In supervised learning, a portion of the sequence of energy levels whose corresponding hidden states are known is required for training a classifier. For comparison purposes, the length of the training sequence is equal to the number of learning slots T_l, as in the case of HMM-training. Two of the techniques studied in this thesis are polynomial classifiers and NARX networks. The following sections explain the procedure with which each of these classifiers is used to determine the availability of a channel.

4.2.1.1. Training a polynomial classifier by minimizing the squared error.

In Section 2.3.2.1, feature expansion was introduced as a means of transforming non-linearly separable data to data that can be separated by a linear hyperplane. The next step is to train the classifier by finding the weight vector $\mathbf{a}$.

One method to find $\mathbf{a}$ is by minimizing the *sum of squared-error* criterion function given by [53]:

$$J(\mathbf{a}) = \|\mathbf{Ya} - \mathbf{b}\|^2 \tag{36}$$

where $\mathbf{Y}$ is a matrix whose rows are the training data, each with feature vector $\mathbf{y}$, and $\mathbf{b}$ is a column vector of positive constants that define the distance of a sample from the separating hyperplane.

The training data $\mathbf{Y}$ is divided into two sets: the training data for class S_1 and the negative of the training data for class S_0.

The solution to (36) is given by [53]:

$$\mathbf{a} = \mathbf{Y}^{\dagger}\mathbf{b}, \tag{37}$$

where $\mathbf{Y}^{\dagger}$ is the pseudoinverse of $\mathbf{Y}$ given by:

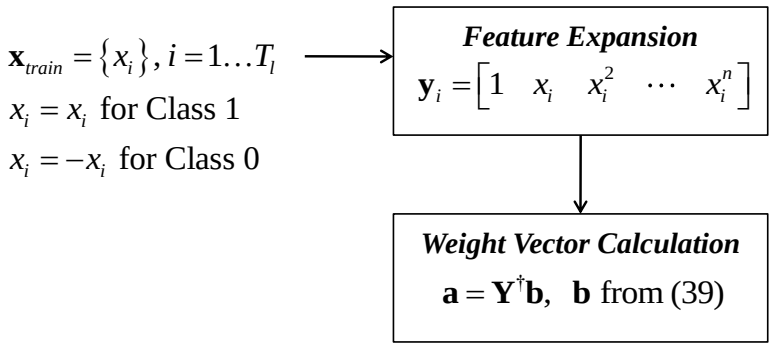
$$\mathbf{Y}^{\dagger} = \left(\mathbf{Y}^t\mathbf{Y}\right)^{-1}\mathbf{Y}^t. \tag{38}$$

It should be noted that the solution given by (37) is not unique. Different values of **b** produce different separating hyperplanes. A common practice is to set **b** as follows [19]:

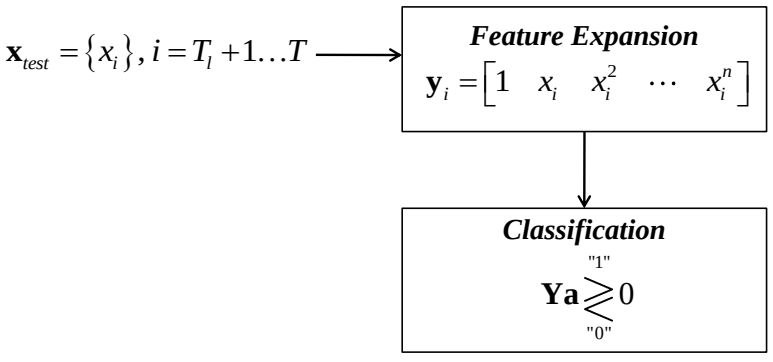
$$\mathbf{b} = \begin{bmatrix} \frac{s}{s_0}\mathbf{1}_0 \\ \frac{s}{s_1}\mathbf{1}_1 \end{bmatrix} \tag{39}$$

where s is the total number of samples, s_0 is the number of samples of class S_0, s_1 is the number of samples of class S_1 and $\mathbf{1}_i$ is a column vector of s_i ones.

The computational complexity of the calculation of the weight vector **a** is $O(T_l\hat{d}^2 + \hat{d}^3 + T_l\hat{d})$, where $\hat{d}$ is dimension of the feature vector after feature expansion. Once the weight vector is computed, the state of the channel is decided by multiplying the expanded feature vector by the weight vector for the rest of the $T - T_l$ timeslots. The decision is made as follows: decide class 1 if score > 0, and class 0 if score < 0. The procedure of polynomial classification is given by the flowchart in Figure 13.



(a) Training phase



(b) Testing phase

Figure 13: Decision on the channel availability by polynomial classifiers: (a) training phase, (b) testing phase

4.2.1.2. Training algorithms for NARX networks. One of the supervised learning techniques is the NARX network introduced in Section 2.3.3. Training a NARX network involves calculation of the synaptic weights and choosing the appropriate activation functions, the number of hidden layers, the number of hidden neurons and the learning algorithm. Several learning algorithms are used to train a NARX network. Regardless of the method of training, the computational complexity analysis of the neural networks is quite involved. In general, the computational complexity of training a neural network is higher than that of polynomial classifiers or the Viterbi algorithm.

One method of training a NARX network is by converting the network into a series-parallel architecture [54]. A series-parallel architecture does not involve a feedback from the output to the input since the output is assumed to be known at the training stage. In this case, a NARX network can be trained as a regular neural network with the static back-propagation algorithm given in Figure 14. The objective function to be minimized is:

$$J(\mathbf{w}) = \frac{1}{2}(t-z)^2 \tag{40}$$

where t is the target output, and z is the output of the neural network, and both can be expressed in terms of the synaptic weights.

Back-Propagation Algorithm

initialize n_H, weight vector $\mathbf{w}$, learning rate η, $m = 0$
repeat
  for k=1 **to** no. of training samples **do**
    $m = m + 1$
    compute the output z_k using (15)
    calculate the error $\Delta\mathbf{w} = -\eta \frac{\partial J}{\partial \mathbf{w}}$
    update the weight vector $\mathbf{w}(m+1) = \mathbf{w}(m) + \Delta\mathbf{w}(m)$
  end for
until all samples correctly classified or other stopping criterion is met
return w

Figure 14: The standard back-propagation algorithm

If the output is not known at the training stage or if the training needs to be done in real-time, another variant of the back-propagation algorithm, known as the *back-propagation-through-time (BPTT)* algorithm is used. The objective function to be minimized is:

$$J(\mathbf{w}(n)) = \frac{1}{2}(t(n) - z(n))^2 \tag{41}$$

Figure 15 shows the BPTT algorithm.

Back-Propagation-Through-Time Algorithm

1: **initialize** n_H, weight vector $\mathbf{w}(n)$, learning rate η
2: **for** n=1 **to** no. of training samples **do**
3: compute the output $z(n)$ using (15)
4: calculate the error $\Delta\mathbf{w}(\mathbf{n}) = -\eta \frac{\partial J(n)}{\partial \mathbf{w}(n)}$
5: update the weight vector $\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n)$
6: **end for**
7: **return** w

Figure 15: The back-propagation-through-time algorithm

Once the NARX network is trained using either of the training algorithms, for the rest of the $T - T_l$ timeslots, the energy of the signal is passed through the network to decide whether the channel is available or not.

In the next chapter, the performance of the classifiers is assessed in terms of the amount of collision with the PUs, and how efficiently the channel was used. The performance of the classifiers will also be tested for various SNRs and under various shadowing conditions.

Chapter 5: Simulation Results

This chapter presents the simulation results that were performed on HMM-generated data simulating the behavior of a PU. Extensive simulations were done to evaluate the performance of the system for different scenarios.

5.1. Simulation Setup

The data used in the simulations were obtained based on a model from real spectrum measurements performed by [1–3]. The measurements in [1] were obtained from the 380-382 MHz band over 25 hours. The measurements were performed at a frequency resolution of 10 kHz and a repetition rate of 4.4 sec. The HMM parameters of the primary user with center frequency 381.7375 MHz and 11 kHz bandwidth were estimated in [1] by the Baum-Welch algorithm in Figure 8, and the results, with a slight modification (discussed shortly), are as follows:

$$\boldsymbol{\pi} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{42}$$

$$\mathbf{A} = \begin{bmatrix} 0.9687 & 0.0313 \\ 0.7970 & 0.2030 \end{bmatrix} \tag{43}$$

$$\boldsymbol{\mu} = \{ -113.9235, \quad -108.4308\} \tag{44}$$

$$\boldsymbol{\Sigma} = \{ 3.4105, \quad 1.5993\} \tag{45}$$

where $\boldsymbol{\pi}$ is the initial state distribution vector with the first and second elements representing the initial probability of being in state 0 (channel available) or 1 (channel busy) respectively, $\mathbf{A}$ is the transition probability matrix, $\boldsymbol{\mu}$ is the mean vector measured in dBm, and $\boldsymbol{\Sigma}$ is the variance vector. The modification made on the results in [1] is the mean of state 1, where the value has been changed from -92.4308 to -108.4308, which corresponds to a lower SNR (6 dB) than the actual spectrum measurements. The change was necessary to evaluate the classification performance under low SNR values. The effect of changing the SNRs is investigated in subsequent sections.

The joint probability density function (pdf) of energy and classes 0 and 1 is shown in Figure 16. As can be seen in the figure, the channel corresponds to an inactive primary user or a channel with low duty cycle.

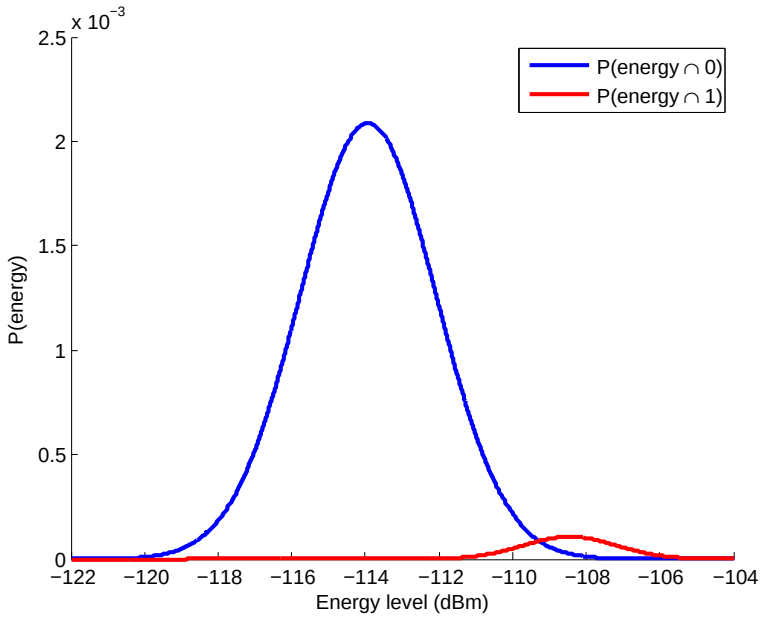


Figure 16: Distribution of the energy levels

Throughout this chapter, the data generation is performed based on the model given by (42)-(45) unless stated otherwise. The software used for the simulations is MATLAB®, with an HMM toolbox distributed under the MIT License [55].

The number of timeslots in each frame was $T = 10000$, out of which the number of the learning timeslots is $T_l = 1000$. As discussed earlier, the length of the timeslot (and hence the length of the learning slots of the frame) depends on the sensing capability of the SU transceiver. If the SU takes a longer time in sensing the channel, the length of the learning vector ρ will decrease, which results in a less accurate estimate of the PU model. Another option is to decrease the number of timeslots the SU gets for transmission. However, this decreases the utilization of the channel. It should be noted that even if the sensing capability of an SU transceiver varies from one SU to another, the assumption that the length of the timeslot is strictly less than the maximum interference time allowed by the PU still holds.

5.2. Classification using Hidden Markov Models

This section discusses the performance of the conventional decoding process of an HMM-generated sequence of observations. Different performance measures, including the collision and utilization probabilities, are evaluated.

5.2.1. Collision probability. The collision probability[1] is defined as the probability with which an SU uses a channel that is occupied by the primary user, given by the ratio

$$collision\ probability = \frac{number\ of\ slots\ collision\ occurs}{total\ number\ of\ slots\ used\ by\ the\ PU}$$

Throughout this chapter, the maximum allowed collision probability is assumed to be 10%.

The collision probability was calculated for 1000 frames and averaged over 10 realizations, as shown in Figure 17.

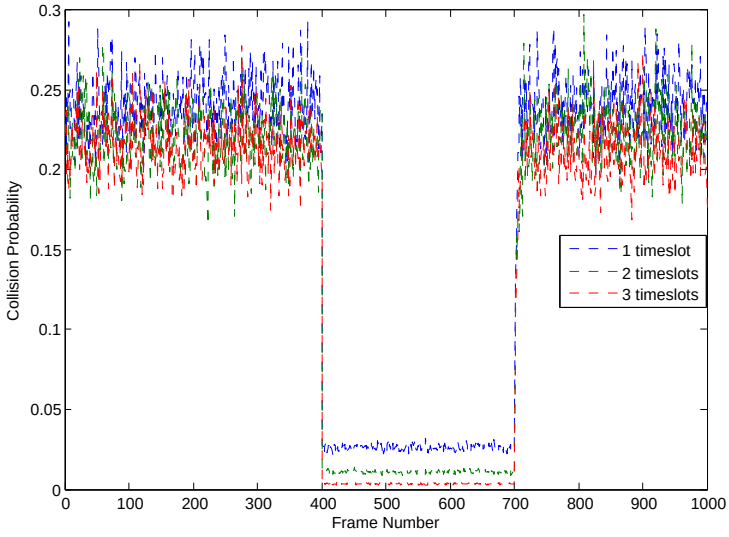


Figure 17: Collision probability over time

In each frame, after the learning phase is completed, the HMM predicts the state of the channel. As proposed in Figure 11, if the current timeslot is free, the SU transmits

[1]The term *collision* in the literature is sometimes also used to refer to the case when SUs try to access a channel simultaneously.

over that timeslot. Other access algorithms were also tested. In [1], the authors propose that two consecutive timeslots should be free for the transmission to be allowed. This requires the SU to wait for one timeslot and then transmit over the next timeslot only if both timeslots were free. The same scenario was repeated with 3 timeslots, where the SU waits for 2 timeslots and transmits over the third only if all the three timeslots are free. With 2 and 3 timeslots, the SU is not allowed to transmit over an available slot, which in turn reduces the utilization of the channel. However, this decreases the probability of making a wrong decision about the availability of the channel, which reduces the probability of collision with the primary user. As can be seen in Figure 17, collision probability decreases as the number of slots increases.

For the first 400 frames and the last 300 frames, the PU was modeled as per (42)-(45). In order to monitor the effect of changing the PU behavior, the transition matrix **A** for the frames between frame numbers 400 and 700 was changed to:

$$\mathbf{A} = \begin{bmatrix} 0.3000 & 0.7000 \\ 0.7970 & 0.2030 \end{bmatrix} \tag{46}$$

The transition matrix was also changed to test the ability of the HMM to track the changes in the PU channel usage pattern, which can be clearly seen in the figure.

The collision probability is highly correlated with the behavior of the primary user. The transition matrix given by (43) corresponds to a channel that is mostly vacant (probability that the channel is vacant and will stay vacant for the next timeslot is 96.87%). The corresponding collision probabilities for the first 400 frames and the last 300 frames are in the range of 18-30%. Such a high collision probability is due to the nature of HMM predictions, which are less accurate when there is no clear pattern of the PU. This is because the predictions are dependent on the previous states of the channel, as explained in Section 4.1.1. Consider a sequence of 1000 '0's followed by a '1'. The probability of the HMM correctly predicting it is a '1' is less than the case where there is an actual pattern of '0's and '1's. Consequently, the SU would assume that a channel is free when in fact it is busy, which leads to a collision. On the other hand, the transition matrix given by (46) corresponds to a more active user and there-

fore, the accuracy of the predictions is much higher, which is translated in low collision probabilities of less than 3% for the frames between 400 and 700.

5.2.2. Utilization probability. The utilization probability is defined as the probability with which an SU uses the channel given that the primary user is not using the channel. Utilization probability is given by the ratio

$$utilization\ probability = \frac{number\ of\ free\ slots\ used\ by\ an\ SU}{total\ number\ of\ free\ slots}$$

Although the proposed access algorithm increases the probability of collision with the primary user, as seen in Section 5.2.1, the utilization probability is increased significantly as seen in Figure 18. The utilization probability increases by at least a factor of 3 for the frames between 400 and 700 when the timeslots were decreased from 2 to 1. This is at the expense of a slight increase in the collision probability. Furthermore, the utilization probability with 1 timeslot stays almost constant, which shows that with the proposed access algorithm, the utilization probability is immune to changes in the PU behavior.

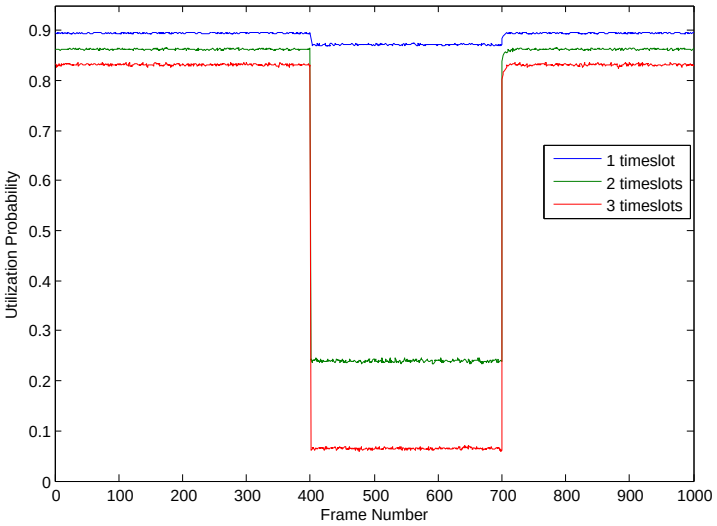


Figure 18: Utilization probability over time

5.2.3. Signal-to-noise ratio (SNR). Another factor that changes the collision probability is the signal-to-noise ratio (SNR). Figure 19 shows the collision probability as the SNR increases from 4 to 12 dB. Since an increase in the SNR corresponds to a higher separation between the means of the 'ON' and 'OFF' distributions, it is easier to decide the state of the primary user and hence the collision probability decreases.

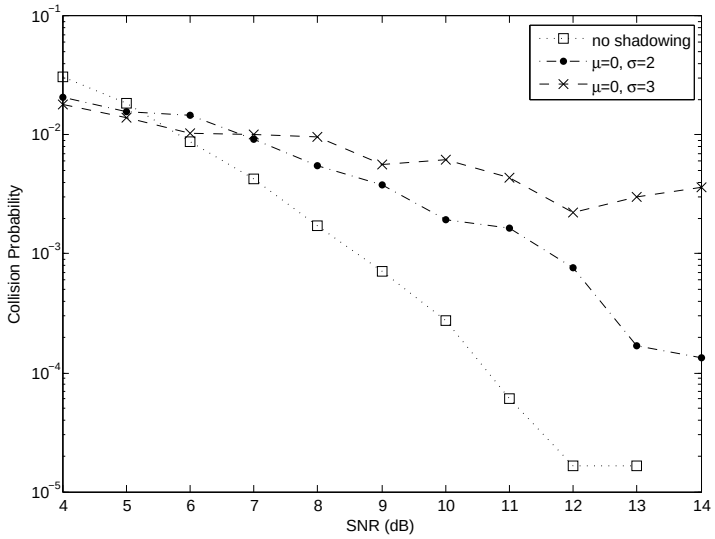


Figure 19: Collision probability vs. SNR

5.2.4. Log-normal shadowing. Log-normal shadowing was introduced by adding a Gaussian random variable ε of zero mean and standard deviation of $\sigma_\varepsilon = 2$ dB and $\sigma_\varepsilon = 3$ dB to the mean of the distribution of state '1' (μ_1), and the results are shown in Figure 19. The higher the value of σ_ε, the higher the probability of collision with the primary user. This is expected because shadowing results in weakening the signal, and the SU decides that the channel is idle when in fact it is busy. According to the impact of shadowing, the SU can wait for three or more consecutive slots before transmission to minimize the collision probability. Another method to mitigate this problem is to have multiple SUs to sense the channel and hence decrease the likelihood of the signal being shadowed.

5.2.5. Learning curve. The HMM takes a few iterations before it correctly estimates the parameters that most likely produced the sequence of observed energy levels of the channel, as explained in Section 4.1.1. The learning curve is defined as the log-likelihood of the estimated parameters, or $\log(P[O|\theta])$. The learning curve at each iteration is shown in Figure 20. It can be seen from the figure that the HMM converges within a few iterations, which is a direct result of using a large training vector of $T_l = 1000$ slots.

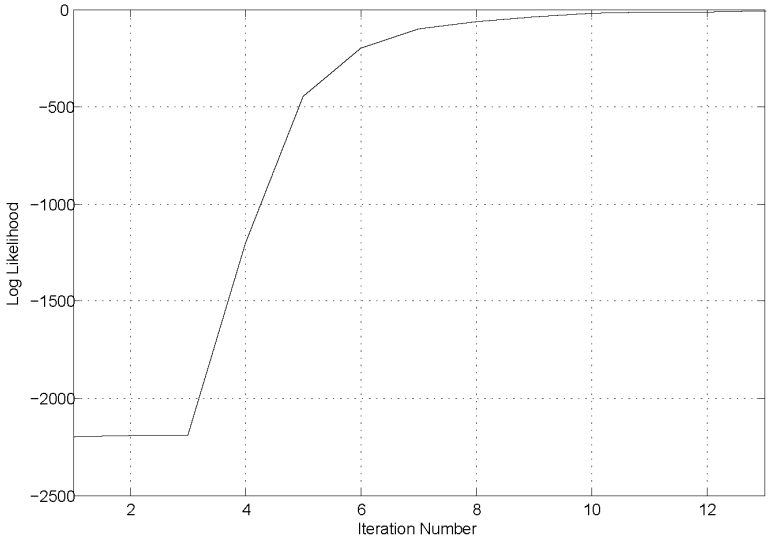


Figure 20: Learning curve of the HMM

5.2.6. Number of learning slots. To monitor the effect of changing the number of learning timeslots (T_l) on the collision probability, various simulations were first performed to determine the minimum number of slots that would guarantee the convergence of the HMM and was found to be $T_l = 600$. The number of learning slots was then varied and examined at different SNR values, and the results are shown in Figure 21.

It can be seen from the figure that at low SNR values, the HMM performance is not governed by the number of learning slots. As long as the HMM converges, the collision probability stays almost the same. On the other hand, at high SNRs, the larger the value of T_l, the lower the collision probability. A 3-D plot is shown in Figure 22 for better visualization.

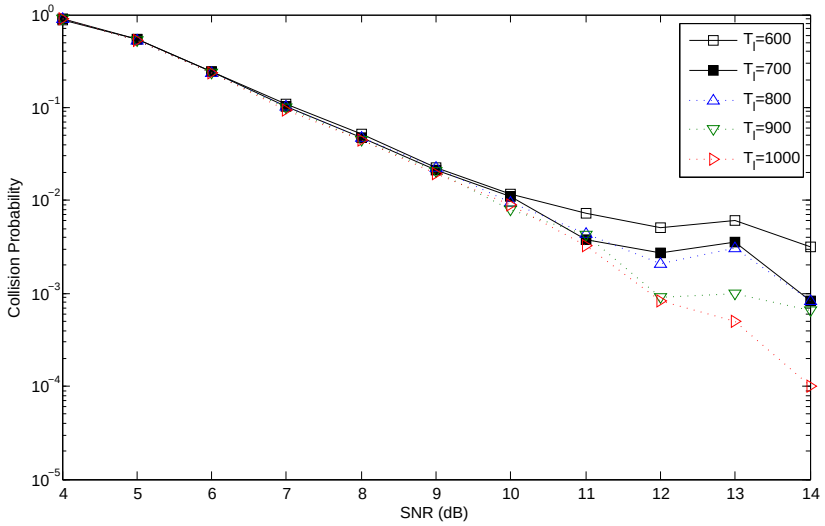


Figure 21: Collision probability vs. SNR

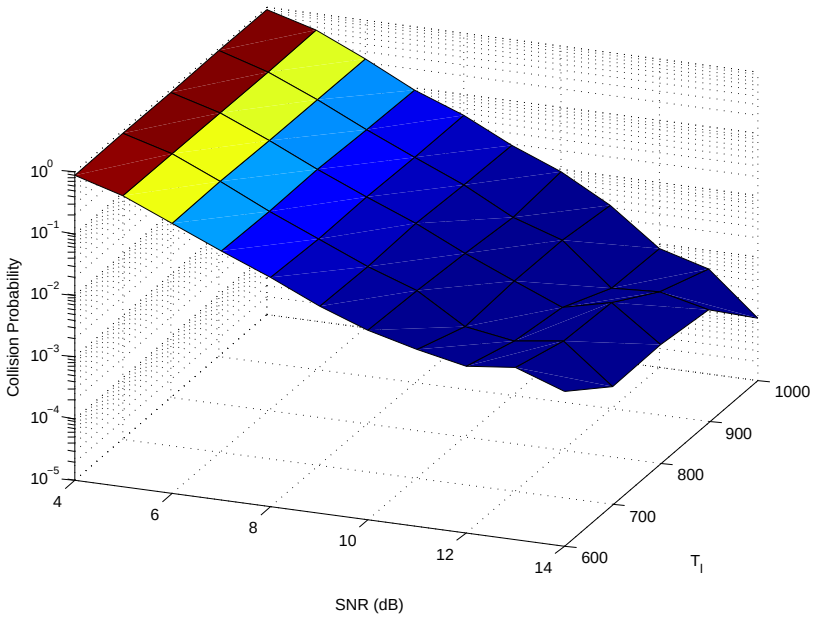


Figure 22: Collision probability vs. SNR vs. T_l

The difference in the performance at high SNRs is due to the fact that the longer the duration of learning, the better the estimation of the channel model and hence the better the prediction of the state of the channel. This cannot be seen at low SNRs since the difference between the means of the 'ON' and 'OFF' distributions is small, and hence the overlap between the two classes is more. Having a longer duration of learning

in this case does not enhance the performance of the HMM since the confusion level of whether the channel is 'ON' or 'OFF' stays the same.

The use of a larger T_l results in a lower collision probability at high SNRs but has a converse effect on the utilization probability as seen in Figure 23.

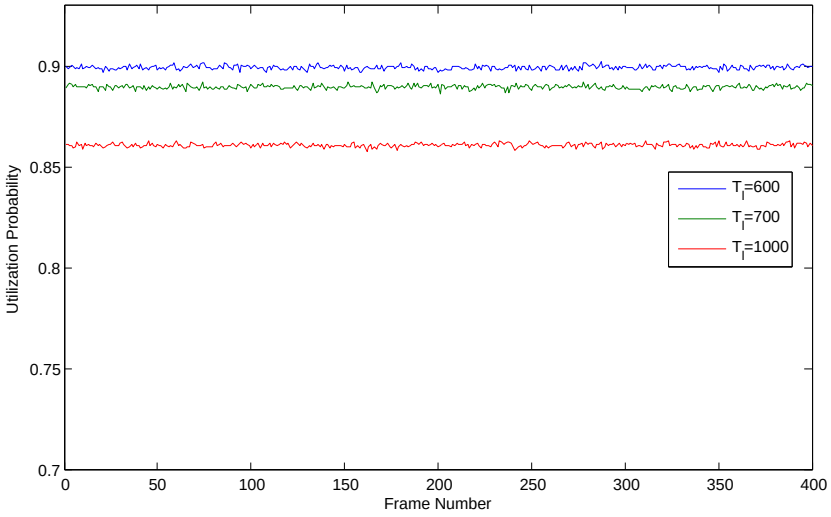


Figure 23: Utilization probability over time

Although the collision probability at higher SNRs is lower with larger T_l, the collision probability at high SNRs is much lower compared to the case of lower SNRs and therefore, the use of a larger T_l is unjustifiable. Furthermore, the utilization probability is improved with a smaller T_l.

5.2.7. Multiple PUs. To study the effect of having multiple PUs, a scenario with 5 PUs was simulated. For every frame, each PU can be in one of 5 activity patterns, given by:

$$\mathbf{A_1} = \begin{bmatrix} 0.9687 & 0.0313 \\ 0.7970 & 0.2030 \end{bmatrix}, \quad \mathbf{A_2} = \begin{bmatrix} 0.3000 & 0.7000 \\ 0.7970 & 0.2030 \end{bmatrix},$$
$$\mathbf{A_3} = \begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix}, \quad \mathbf{A_4} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}, \quad \mathbf{A_5} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{47}$$

Clearly, the transition matrix $\mathbf{A_1}$ corresponds to the least active primary user, followed by $\mathbf{A_3}$, and the transition matrices $\mathbf{A_2}$, $\mathbf{A_4}$, and $\mathbf{A_5}$ correspond to higher activities and result in less number of free timeslots. Moreover, these transition matrices are assumed to be equally likely.

The collision and utilization probabilities over time are shown in Figure 24 and Figure 25 respectively.

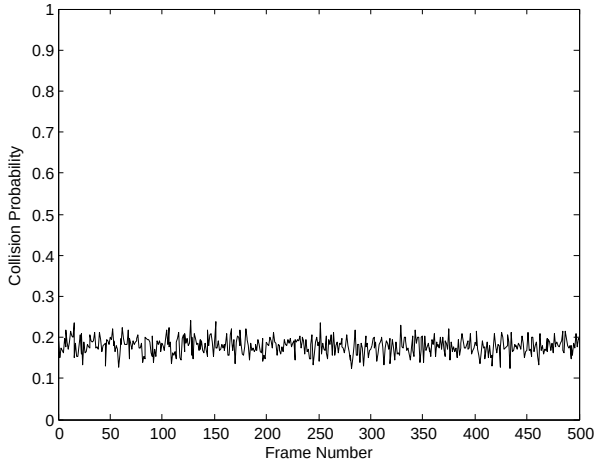


Figure 24: Collision probability over time with 5 PUs

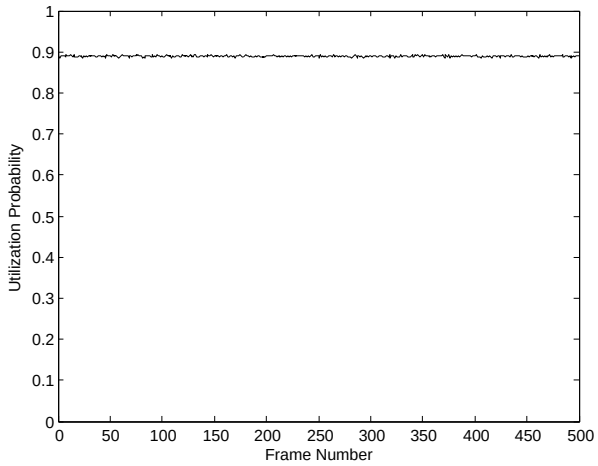


Figure 25: Utilization probability over time with 5 PUs

It can be seen in the figures that the pattern does not change much, which suggests that the SU always chooses the channel with the least activity. Moreover, the probability that none of the channels has the transition matrix $\mathbf{A_1}$ or $\mathbf{A_3}$ is low ($0.6^5 = 0.08$). Therefore, in contrast to the case of 1 PU simulated in Sections 5.2.1-5.2.2, as the number of PU increases, the changes in the pattern of the collision and utilization probabilities decreases.

5.3. Classification using Polynomial Classifiers

In this section, the data generated using the model (42)-(45) is used to train a polynomial classifier. As in the case with HMM-training, the number of learning slots is $T_l = 1000$ and the total number of frames $T = 10000$. For the case of polynomial classifiers, the state of the channel for each of the learning timeslots is assumed to be known by the SU. This assumption is valid if the learning slots are considered as pilot signals sent by the SU base station periodically, and are treated as transmission overhead.

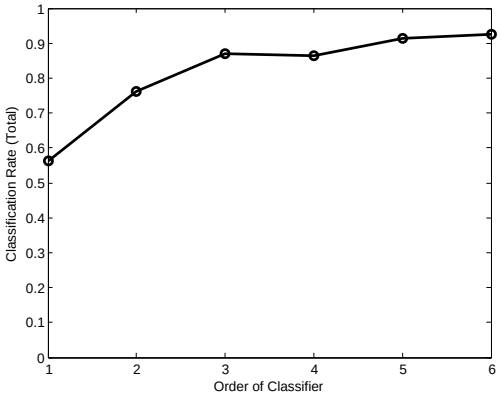
At the end of the training phase of each frame, a training vector ρ is formed. The weight vector $\mathbf{a}$ is then calculated by the procedure outlined in Section 4.2.1.1. For the the rest of the $T - T_l$ timeslots, the expanded feature vector is multiplied by the weights. If the score is greater than zero, the state of the channel is classified as '1', otherwise it is classified as '0'.

In the following subsections, the performance of the classifiers is studied thoroughly. In all the following simulations, the results are averaged over 400 realizations.

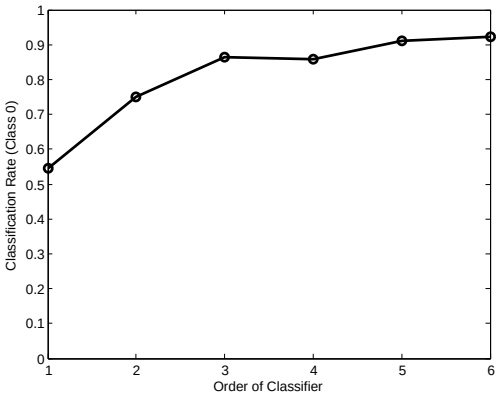
5.3.1. Classification accuracy. The performance of the polynomial classifier is evaluated in terms of the overall classification rate, the classification rate of class 0 and the classification rate of class 1. The overall classification rate is defined as the fraction of correctly classified states to the total number of states. The classification rate of class 0 is the fraction of correctly classified class 0 states to the total number of class 0 states. Since class 0 represents an available channel, the higher the classification rate of class 0, the higher the spectrum efficiency. Similarly, the classification rate of class 1 is the fraction of correctly classified class 1 states to the total number of class

1 states. The higher the classification rate of class 1, the less likely an SU will collide with a PU.

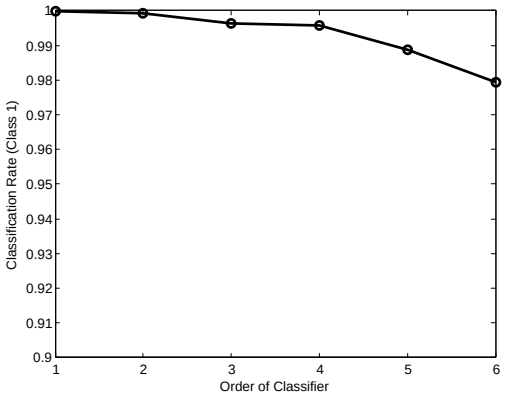
The order of the polynomial classifier was varied from $n = 1$ to $n = 6$ to observe the effect of the polynomial order on the classification rate. Figure 26 shows the classification rate vs. the order of feature expansion. As can be seen in the figure, the overall classification accuracy increases with the order of the classifier, which suggests that the two classes are non-linearly separable.



(a) Overall classification accuracy



(b) Class 0 classification accuracy



(c) Class 1 classification accuracy

Figure 26: Classification accuracy vs. order of classifier

5.3.2. Collision and utilization probabilities. The collision probability was calculated for different orders of the classifier and is shown in Figure 27.

By definition, the collision probability is given by:

$$\textit{collision probability} = 1 - \textit{classification rate of class } \text{`1'}.$$

The collision probability increases slightly with the order of the classifier. This is because the number of class '1' samples are much less that class '0' samples, as seen in Figure 16, which leads to the classifier being more sensitive to errors in the classification rate of class '1'. However, even with a polynomial of order $n = 6$, the collision probability is less than 2%, which is quite acceptable.

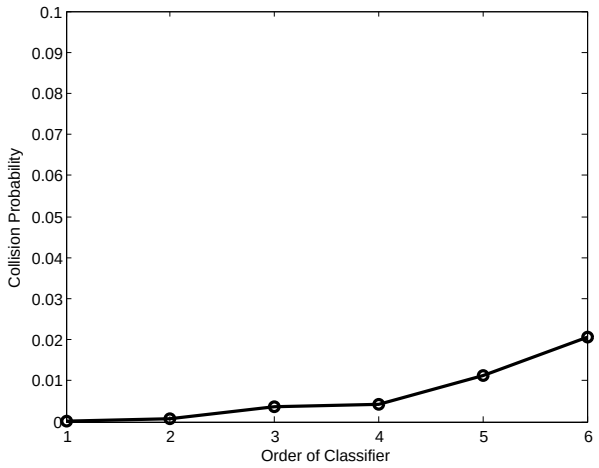


Figure 27: Collision probability vs. order of classifier

The utilization probability was also calculated for different orders of the classifier and is shown in Figure 28. By definition, the utilization probability is essentially the classification rate of class '0'. The utilization probability increases significantly with the order of the classifier. This gain in the performance comes at an expense of higher computational complexity. In particular, the computational complexity of the calculation of the weight vector $\mathbf{a}$ is $O(T_l d^2 + d^3 + T_l d)$, where T_l is the length of the training vector ρ, and d is the dimension of the feature vector ($d = n + 1$). Moreover, there is diminishing gain in the performance after order 3.

5.3.3. Effect of changing the SNR. The collision and utilization probabilities are calculated for various SNRs with $n = 1$, and the result is shown in Figure 29.

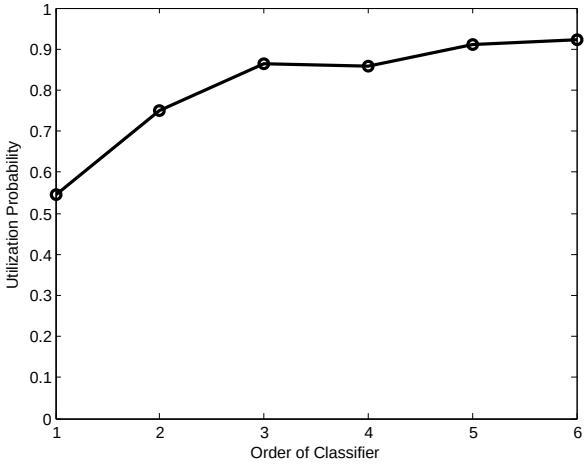


Figure 28: Utilization probability vs. order of classifier

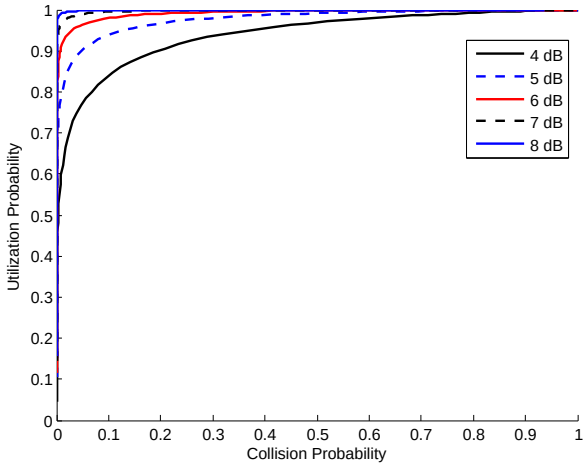


Figure 29: Utilization probability vs. collision probability for various SNRs

As the SNR increases, the utilization increases for the same collision probability. This is expected because a higher SNR corresponds to a higher separation between the means of the 'ON' and 'OFF' distributions and hence better classification performance.

5.3.4. Comparison with other existing energy-detection schemes. The performance using polynomial classifiers was compared to other schemes, namely the conventional energy detection (ED) described in Section 3.3, and a modified ED scheme proposed by [38]. It is worth noting that the comparison is approximate since the PU model adopted by the proposed scheme (the HMM model) is different from the other schemes. Moreover, the terminologies used in the performance evaluation of spectrum

sensing and access schemes are different. In spectrum sensing, the performance of the system is assessed in terms of the probability of false alarm and the probability of detection. The probability of false alarm is defined as the probability that the signal level is above the threshold γ given that the channel is available. The complement of the probability of false alarm is analogous to the utilization probability, and hence, and with a slight abuse of notation, the utilization probability can be expressed in terms of the probability of false alarm as:

$$\textit{utilization probability} = 1 - \textit{probability of false alarm}.$$

Similarly, the probability of detection is defined as the probability that the signal level is above the threshold γ given that the channel is busy. The collision probability can be expressed in terms of the complement of the probability of detection, the probability of miss-detection as:

$$\textit{collision probability} = \textit{probability of miss-detection}.$$

Figure 30 shows the utilization vs. the collision probabilities for the two ED schemes and the polynomial classifier of orders n=1, 2,& 3.

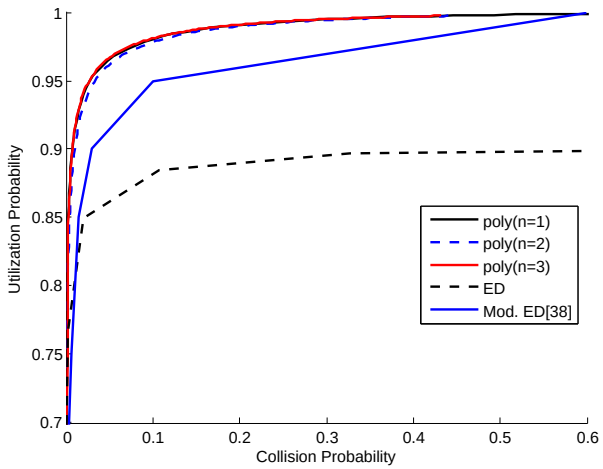


Figure 30: Comparison with other existing schemes

It can be seen in the figure that the polynomial classifier outperforms the other two schemes. The polynomial classifiers with different orders, however, have a comparable performance in terms of utilization versus collision probabilities. The advantages of having higher-order polynomials will be prominent in subsequent sections.

5.3.5. Multiple-timeslot scenario. So far, the feature vector (before expanding the features) was of dimension ($d = 1$) since the energy level of the current timeslot only was taken into consideration. The feature vector x could be expanded to incorporate energy levels from previous timeslots. For example, if measured energy levels of the previous two timeslots need to be taken into account, the feature vector will be:

$$\mathbf{x} = \begin{bmatrix} x_{t-2} & x_{t-1} & x_t \end{bmatrix}$$

This introduces delay to the system since the SU has to wait for a couple of timeslots before making a decision on the availability of the channel. However, better performance is expected if the data is correlated.

The classification rate results for classes '1' and '0' will be omitted in the following discussions since they are directly related to the collision and utilization respectively. Figures 31, 32 and 33 show the classification rate, collision probability and utilization probability respectively, as previously explained. The results suggest that the data is uncorrelated since the performance is degraded with increasing delays.

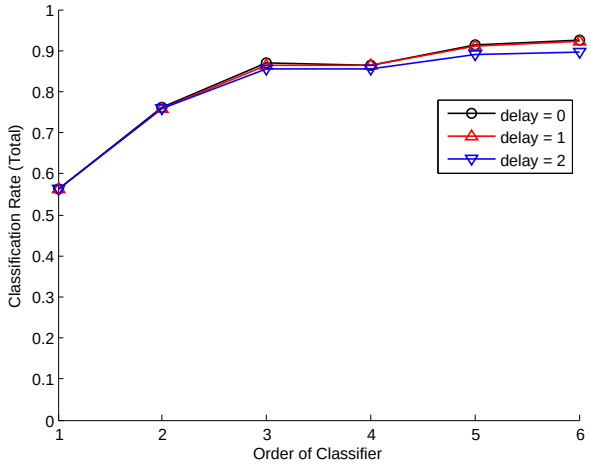


Figure 31: Classification accuracy vs. order of classifier for the multiple-timeslot scenario

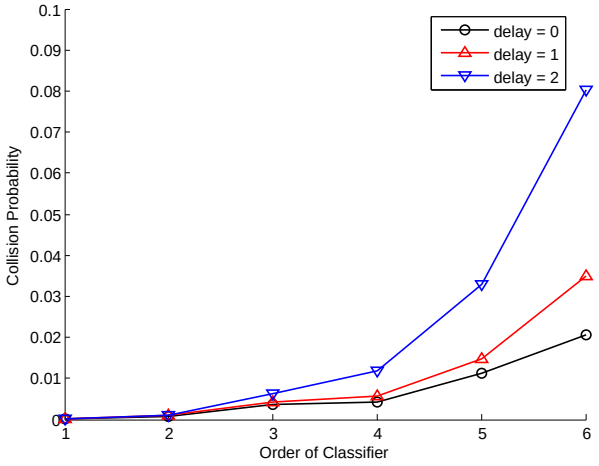


Figure 32: Collision probability vs. order of classifier for the multiple-timeslot scenario

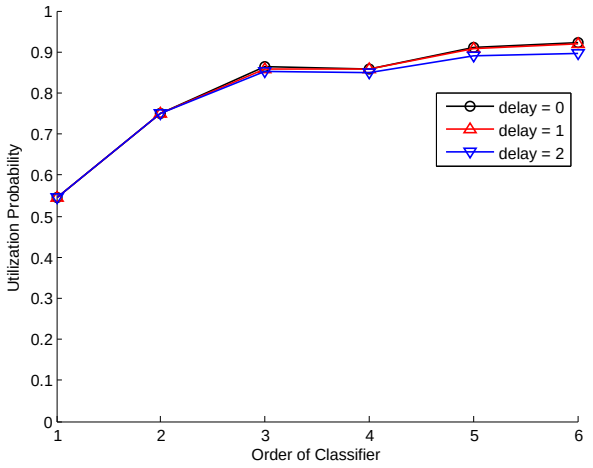


Figure 33: Utilization probability vs. order of classifier for the multiple-timeslot scenario

To introduce correlation to the data, the transition matrix **A** was changed to:

$$\mathbf{A} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix} \tag{48}$$

where a_{00} and a_{11} were set to 0.8, which corresponds to a higher correlation in the '0' and '1' states.

The results are shown in Figures 34-36. It can be seen from the figures that there is a negligible improvement in the performance with delay=1 and delay=2 over the no delay case since the data is correlated. Furthermore, the Markov property states that a state is dependent on the previous state, which is reflected in the figures where the best performance is when delay=1.

Although there is a slight improvement in the sensing accuracy, the use of delays is unjustifiable for the following reasons: (1) delays improve the performance only if the data is correlated, which is not always the case as seen in the results, (2) the computational complexity increases exponentially with increasing delay, and (3) the performance gain is insignificant. Hence, the use of delays should not be considered.

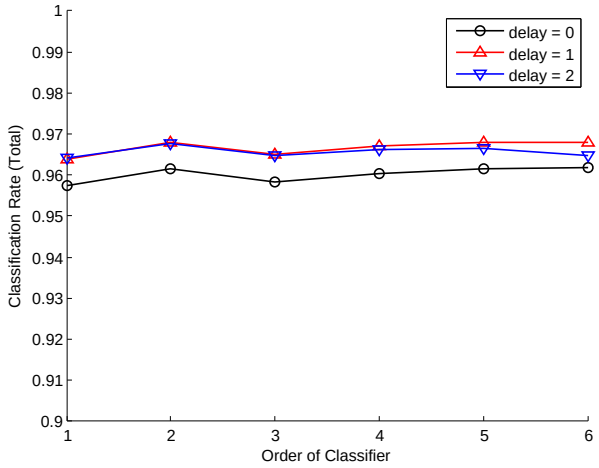


Figure 34: Classification accuracy vs. order of classifier with correlated data

5.3.6. Multiple-antenna scenario. The performance of the system is expected to improve if the SU has more than one antenna. Although the antennas are physically close to each other, each antenna will receive a different signal and hence diversity gain is achieved. This comes at the price of computational complexity, which increases exponentially with increasing number of antennas and higher orders of the classifier.

For the simulation of the multiple-antenna scenario, the antennas are correlated with a correlation coefficient of 0.7.

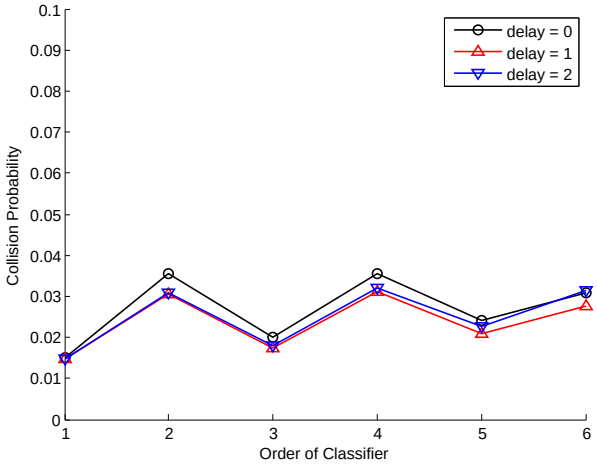


Figure 35: Collision probability vs. order of classifier with correlated data

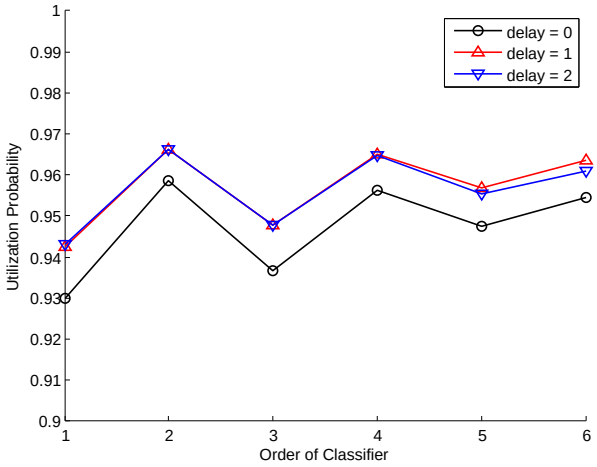


Figure 36: Utilization probability vs. order of classifier with correlated data

Figures 37-39 show the classification rate, collision probability and utilization probabilities respectively. As seen in the figures, the enhancement in the performance is minimal, and hence, using multiple antennas in this case will only increase the computational complexity without any significant gain. This is mainly because small-scale fading was not taken into account. It is expected that with small-scale fading, the improvement in the system performance with multiple antennas should be much higher.

5.3.7. Log-normal shadowing. To monitor the performance of the system under the effect of log-normal shadowing, a random variable ε is added to the mean of the 'ON' distribution.

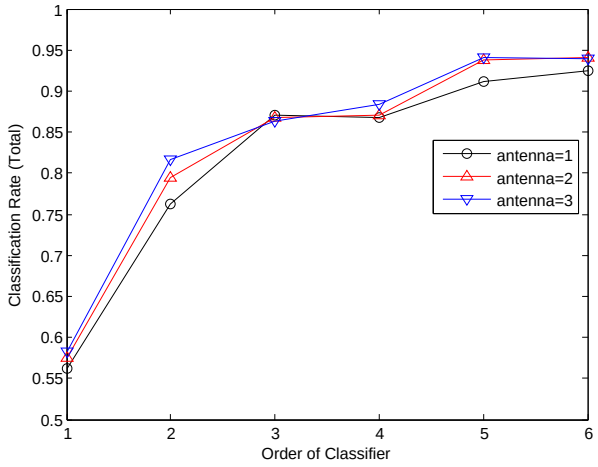


Figure 37: Classification accuracy vs. order of classifier for the multiple-antenna scenario

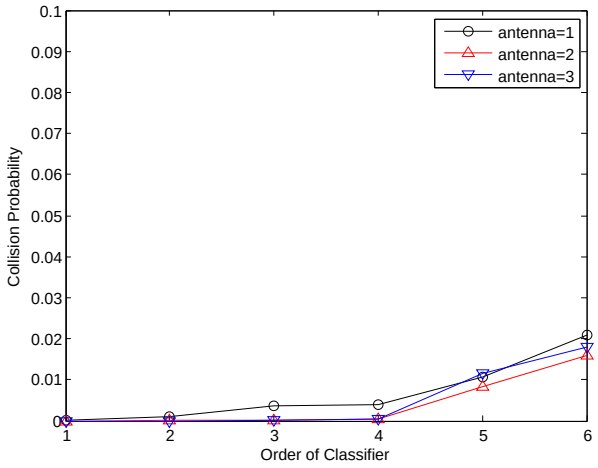


Figure 38: Collision probability vs. order of classifier for the multiple-antenna scenario

The classification accuracy and the collision and utilization probabilities were calculated for different levels of shadowing using polynomial classifiers of order $n = 1, 3 \& 6$, and are shown in Figures 40-42.

At polynomial of order $n = 1$, the utilization probability stays almost constant at around 55% for increasing values of σ_ε, whereas the collision probability increases significantly from close to 0% to 30%. This is expected since the a larger σ_ε corresponds to a larger overlap between the 'ON' and 'OFF' distributions, and hence introduces more errors.

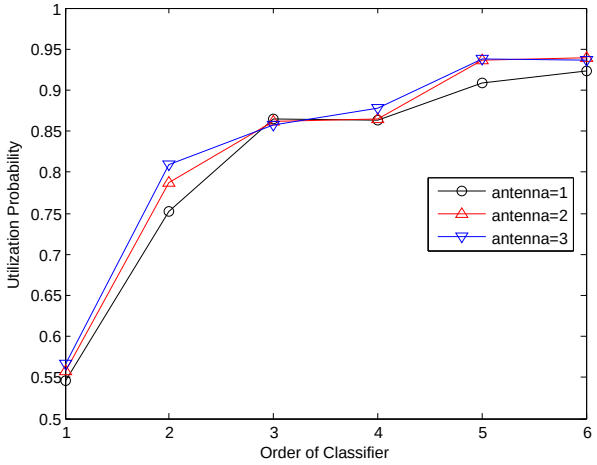


Figure 39: Utilization probability vs. order of classifier for the multiple-antenna scenario

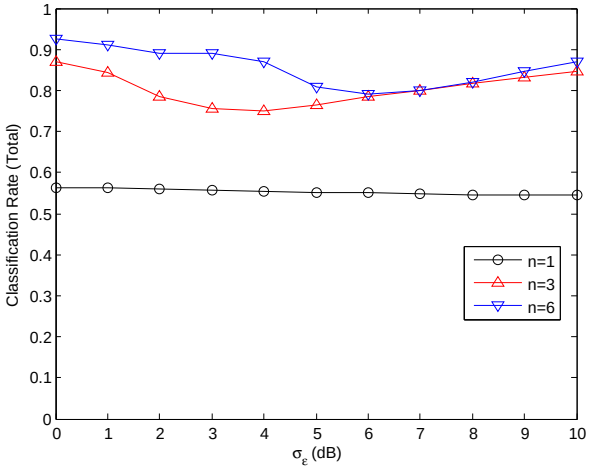


Figure 40: Classification accuracy vs. σ_ε

In addition, since the channel is mostly idle, the number of '1's is much less than the number of '0's, causing the system to be more sensitive to misclassification errors of state '1' and hence a higher collision probability.

It is also noted that, as discussed in previous sections, the utilization probability increases significantly with increasing order of the classifier, at the cost of a slight increase in the collision probability. What is of more interest is the behavior of higher order classifiers to larger shadowing. The results suggest that after a certain limit, as shadowing is increased, collision probability is reduced. Hence, although a polynomial of order $n = 6$ increases the computational complexity, the collision probability

decreases significantly. The utilization probability, however, is still slightly affected by the high shadowing.

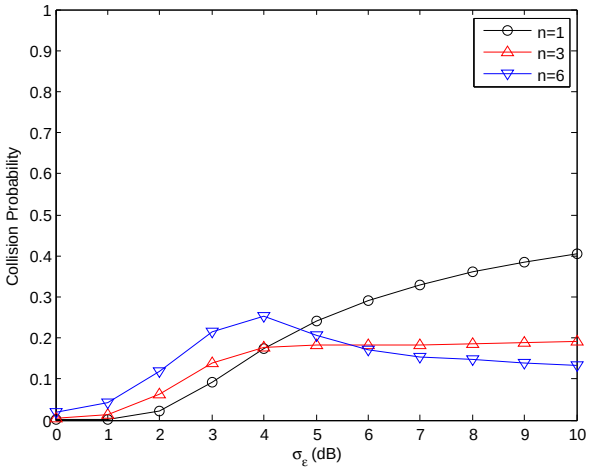


Figure 41: Collision probability vs. σ_ε

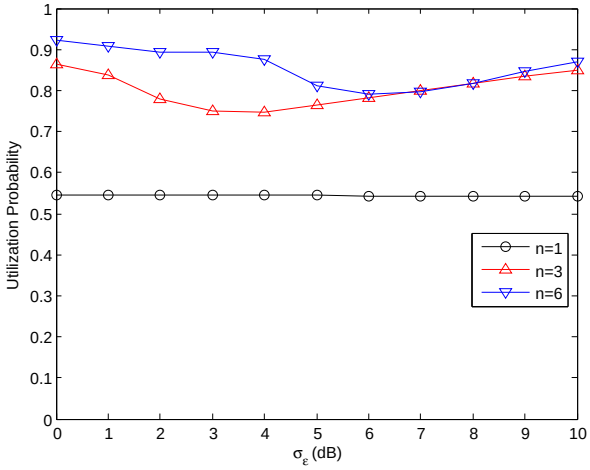


Figure 42: Utilization probability vs. σ_ε

5.3.8. Cooperative spectrum sensing. To mitigate the effect of shadowing, multiple SUs could cooperate to form a joint decision. The results are sent to a fusion center that decides on the availability of the channel. The decision can be made in various techniques, some of which are:

- *majority vote*- each SU decides whether the channel is available or not. The results are then sent to the fusion center which makes a decision based on the majority of the votes.
- *fusion at the feature level*- a feature vector that comprises of the energy level from each user is formed at the fusion center. The feature vector is then expanded, and the weight vector is calculated. At the testing phase, the energies from users are multiplied by the weight vector, and the decision is made based on the sign of score.
- *fusion at the score level*- training of the polynomial classifier is done by each SU individually. At the testing phase, the energy measured by each SU is multiplied by its corresponding weight vector and a score is obtained. The fusion center then makes a decision based on the weighted average of the scores. In the following simulations the scores are weighted based on their sign. If the score is positive (corresponding to a busy channel), the score is divided by s/s_1, where s is the total number of training samples, and s_1 is the number of samples of class 1. Similarly, if the score is negative, the score is divided by s/s_0, where s_0 is the number of samples of class 0. This is done because the score should ideally be either s/s_0, or s/s_1, as defined in (39).

Three SUs under different SNR conditions were simulated in this scenario. All SUs experience shadowing of $\sigma_\varepsilon = 2$ dB. The results are shown in Figures 43−45.

Regardless of the method of data fusion, having multiple users enhances the classification accuracy. There is no clear trend in the results, but in general, fusion of scores outperforms the rest of the schemes in terms of collision and utilization probabilities at all classifier orders,with an exception of $n = 3$. At lower orders of the classifier, feature fusion and score fusion have the same performance in terms of collision probability, however utilization probability increases by about 10% when score fusion is used. It seems that the cross-terms introduced by expanding a feature vector with 3 elements do not help in classification. In fact, it can be seen from the figures that, especially at higher orders of the classifier, the performance using feature fusion is even worse than the majority vote.

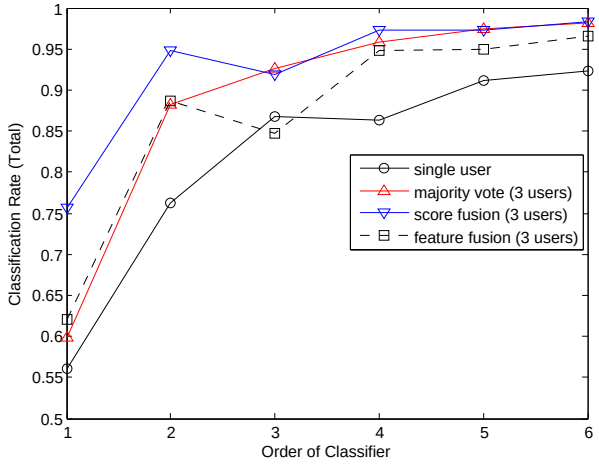


Figure 43: Classification accuracy vs. order of classifier for the multiple-SU scenario

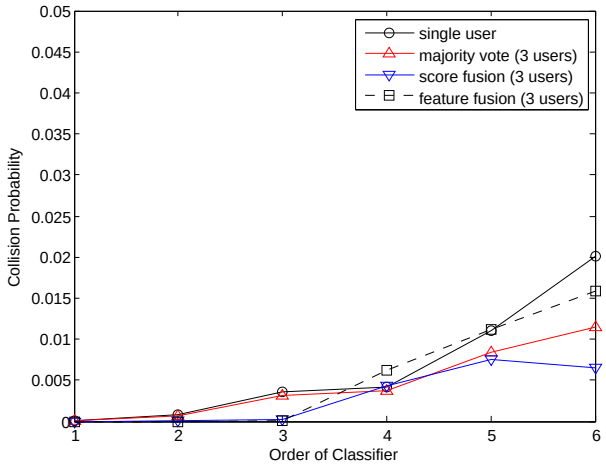


Figure 44: Collision probability vs. order of classifier for the multiple-SU scenario

5.4. Classification using NARX

As for the case with polynomial classifiers, the data is generated using an HMM and is trained using a nonlinear autoregressive with exogenous inputs (NARX) model network. The NARX network was trained using the two algorithms in Section 4.2.1.2. It was found that the back-propagation algorithm performs better than back-propagation-through-time algorithm in terms of classification accuracy since the former uses supervised learning. Different parameters of the NARX network were changed to achieve the best performance as follows:

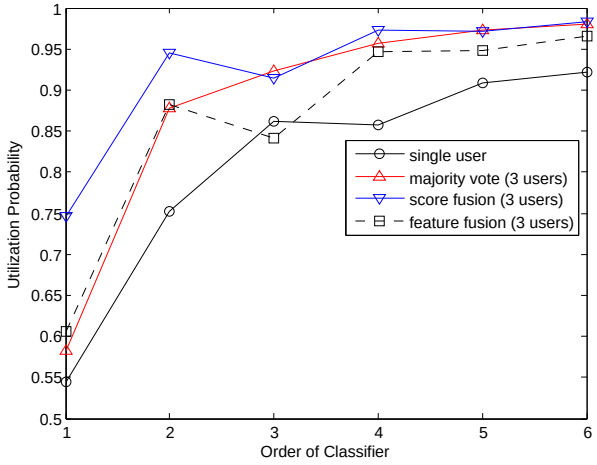


Figure 45: Utilization probability vs. order of classifier for the multiple-SU scenario

- *Input delays*- the number of input delays was changed with delay=1 and delay=2. Better classification performance was obtained with delay=1.
- *Output delays*- similar to input delays, 1 and 2 output delays were tested and a higher classification rate was achieved with delay=2.
- *Number of hidden layers and neurons*- it is a common practice to have only one hidden layer and multiple neurons rather than having multiple hidden layers. With hidden layers=1, the number of neurons were changed from 1 to 5, and the optimal number was found to be 2.
- *Activation function*- different activation functions of the neurons in the hidden layer were used, including the tan-sigmoid function, which generates outputs between -1 and 1, and the log-sigmoid function, which generates outputs in the range of 0 to 1. The function that results in better performance was found to be the tan-sigmoid function.

Table 1 lists the optimal parameters for training, and the results are shown in Figure 46. It can be seen from the figure that the curves overlap at the region for collision probabilities higher than 10%, implying that the performance of the system is not governed by the SNR, which is not generally true. Moreover, even at low probabilities, the utilization performance is worse than both HMMs and polynomial classifiers. Therefore, the use of NARX is not recommended.

Table 1: Parameters of the NARX network

Parameter	***Value***
Input delays q	1
Output delays	2
Number of hidden layers	1
Number of hidden neurons	2
Activation function of hidden layers	tan-sigmoid

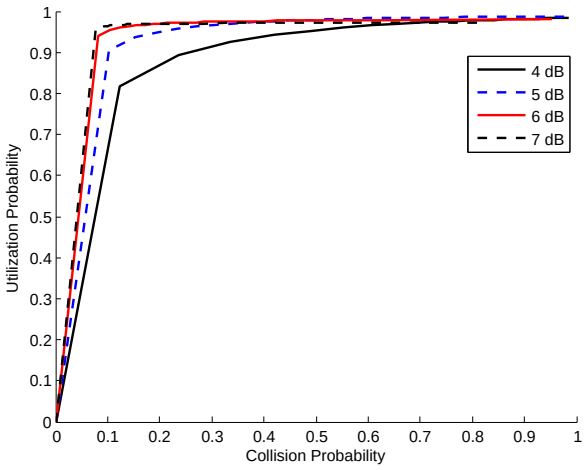


Figure 46: Utilization probability vs. collision probability for the NARX scheme

5.5. Comparison of the Learning Schemes

This section compares between the different learning techniques that were used to determine the availability of the PUs. Classification using NARX is excluded from the comparison since it was found to produce inconsistent results.

The main reason behind the superior performance of polynomial classifiers over the Viterbi algorithm is because the former uses labeled samples for training, or more formally, supervised learning. The collision probability using the Viterbi algorithm is around 25%, which is not acceptable. The collision probability improves by a factor of 25 when using polynomial classifiers. In terms of utilization probability, the performance of the two schemes are comparable. The utilization probability with the Viterbi algorithm is around 90%, which is achieved by the polynomial classifiers only at higher orders of the classifier. Moreover, in terms of the performance under shadowing, polynomial classifiers exhibit a unique behavior where the collision probability decreases

with increasing levels of shadowing. Finally, polynomial classifiers outperform the Viterbi algorithm in terms of computational complexity.

Table 2 summarizes the main differences in the system performance with polynomial classifiers and the conventional HMM-decoding approach, the Viterbi algorithm.

Table 2: Comparison of the learning schemes for spectrum sensing

	Viterbi Algorithm	***Polynomial Classifier***
Training mode	Unsupervised	Supervised
Collision probability	25%	$< 2\%$
Utilization probability	90%	55-92%
Shadowing	$\sigma_\varepsilon \uparrow$ col. prob.$\uparrow$	$\sigma_\varepsilon \uparrow$ col. prob.$\downarrow$
Computational complexity	High	Low

Chapter 6: Conclusions

Cognitive radio users are required to minimize the interference caused to the primary users such that SU transmissions do not affect PU transmissions. In order to do so, an access scheme is proposed where the secondary user is not allowed to transmit unless the channel is free for the current timeslot. As the number of slots increases, the collision probability decreases, but the utilization probability also decreases. Transmission based on one slot was proven to increase the utilization probability, which reaches up to more than a factor 10 in some cases. This is at the expense of a slightly higher collision probability. Results show that with the conventional HMM-decoding approach for spectrum sensing (the Viterbi algorithm), the collision probability exceeds 20%, which is not acceptable.

To reduce the collision probability, the use of polynomial classifiers has been proposed, where a feature vector containing the energy sensed by the SU is expanded to higher orders in order to make the data linearly separable. Results show that using polynomial classifiers achieves considerably lower collision probabilities ($< 1\%$) while maintaining high utilization probabilities. Moreover, several other cases exhibit an improved performance with the use of polynomial classifiers. In particular, higher order polynomial classifiers show significant enhancement in the performance in the case of extreme shadowing. Furthermore, the performance of the system is enhanced in terms of computational complexity, compared to the Viterbi algorithm.

The case of multiple secondary users was briefly discussed, where a cooperative sensing scheme was adopted. Different methods of data fusion were tested, and the results show that the highest classification rates are obtained when the fusion is done at a score level.

Another supervised learning technique, namely NARX networks, was tested with the aim of achieving higher classification rates. However, the use of NARX networks for the decision on the availability of primary users is ruled out due to the inexplicable randomness in the obtained results. Moreover, NARX is more computationally complex than both the Viterbi algorithm and polynomial classifiers.

For future work, the polynomial classifiers could be tested for different patterns of primary users, to investigate whether or not the gain in the performance is only shown in the cases where the primary user is inactive. Moreover, the case of cooperative sensing could be further studied in terms of the collisions between SUs and the possible scheduling schemes that reduce these collisions.

References

[1] T. Nguyeny, B. Mark, and Y. Ephraim, "Hidden Markov process based dynamic spectrum access for cognitive radio," in *2011 45th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2011, pp. 1–6.

[2] T. Nguyen, "Hidden Markov model based spectrum sensing for cognitive radio," Ph.D. dissertation, George Mason University, Fairfax, VA, May 2013.

[3] T. Nguyen, B. Mark, and Y. Ephraim, "Spectrum sensing using a hidden bivariate Markov model," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4582–4591, Sep. 2013.

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

[5] M. Calabrese, "The end of spectrum scarcity: Building on the TV bands database to access unused public airwaves," New America Foundation, Tech. Rep. 9, Jun. 2009.

[6] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Institute of Technology, Sweden, May 2000.

[7] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[8] I. F. Akyildiz *et al.*, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006.

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

[10] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. NewYork, NY: McGraw-Hill, 2014.

[11] E. Hossein, D. Niyato, and Z. Han, *Dynamic Spectrum Access and Management in Cognitive Radio Networks*. Cambridge, UK: Cambridge University Press, 2009.

[12] K. W. Choi and E. Hossain, "Opportunistic access to spectrum holes between packet bursts: A learning-based approach," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2497–2509, Aug. 2011.

[13] A. Goldsmith *et al.*, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, May 2009.

[14] I. Akyildiz *et al.*, "A survey on spectrum management in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 40–48, Apr. 2008.

[15] Y. Saleem and M. H. Rehmani, "Primary radio user activity models for cognitive radio networks: A survey," *Journal of Network and Computer Applications*, vol. 43, pp. 1 – 16, Aug. 2014.

[16] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, 3rd ed. Prentice Hall, 2008.

[17] S. Bayhan and F. Alagz, "Distributed channel selection in CRAHNs: A non-selfish scheme for mitigating spectrum fragmentation," *Ad Hoc Networks*, vol. 10, no. 5, pp. 774 – 788, 2012.

[18] L. Melin-Gutirrez *et al.*, "HF spectrum activity prediction model based on HMM for cognitive radio applications," *Physical Communication*, vol. 9, pp. 199 – 211, Dec. 2013.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[20] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[21] M. Gales and S. Young, *The Application of Hidden Markov Models in Speech Recognition*. Hanover, MA: Now Publishers, 2008.

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[23] D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms,Architectures and Stability*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[24] P. Di Lorenzo, S. Barbarossa, and A. Sayed, "Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3183–3197, Jun. 2013.

[25] M. Tang *et al.*, "Nonconvex dynamic spectrum allocation for cognitive radio networks via particle swarm optimization and simulated annealing," *Computer Networks*, vol. 56, no. 11, pp. 2690 – 2699, 2012.

[26] Z. Zhang *et al.*, "On swarm intelligence inspired self-organized networking: Its bionic mechanisms, designing principles and optimization approaches," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 513–537, 2014.

[27] B. Atakan and O. Akan, "Biologically-inspired spectrum sharing in cognitive radio networks," in *2007 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2007, pp. 43–48.

[28] G. Li *et al.*, "Enhanced biologically-inspired spectrum sharing for cognitive radio networks," in *2010 IEEE International Conference on Communication Systems (ICCS)*, Nov. 2010, pp. 767–771.

[29] A. Martnez-Vargas and ngel G. Andrade, "Comparing particle swarm optimization variants for a cognitive radio network," *Applied Soft Computing*, vol. 13, no. 2, pp. 1222 – 1234, 2013.

[30] Z. Zhang, K. Long, and J. Wang, "Self-organization paradigms and optimization approaches for cognitive radio technologies: a survey," *IEEE Wireless Communications Magazine*, vol. 20, no. 2, pp. 36–42, Apr. 2013.

[31] C. Do *et al.*, "A lightweight algorithm for probability-based spectrum decision scheme in multiple channels cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 3, pp. 509–512, Mar. 2013.

[32] S. Stotas and A. Nallanathan, "On the throughput and spectrum sensing enhancement of opportunistic spectrum access cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 97–107, Jan. 2012.

[33] D. Niyato, E. Hossain, and Z. Han, "Dynamic spectrum access in IEEE 802.22-based cognitive wireless networks: a game theoretic model for competitive spectrum bidding and pricing," *IEEE Wireless Communications Magazine*, vol. 16, no. 2, pp. 16–23, Apr. 2009.

[34] M. Tehrani and M. Uysal, "Auction based spectrum trading for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 6, pp. 1168–1171, Jun. 2013.

[35] L. Cao and H. Zheng, "Distributed rule-regulated spectrum sharing," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 130–145, Jan. 2008.

[36] S.-S. Tan, J. Zeidler, and B. Rao, "Opportunistic spectrum access for cognitive radio networks with multiple secondary users," *IEEE Transactions on Wireless Communications*, vol. 12, no. 12, pp. 6214–6227, Dec. 2013.

[37] Z. Zhang *et al.*, "Channel exploration and exploitation with imperfect spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 3, pp. 429–441, Mar. 2013.

[38] I. Sobron *et al.*, "Energy detection technique for adaptive spectrum sensing," *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 617–627, Mar. 2015.

[39] M. Fahim, M. Ismail, and H. Tawfik, "An innovative cooperative spectrum sensing algorithm with non-ideal feedback channels and delay considerations," *Wireless Personal Communications*, vol. 78, no. 1, pp. 313–332, 2014.

[40] M. Fahim, M. Ismail, and H. Tawfik, "Multi-stage energy detection spectrum sensing using group intelligence," in *2013 IEEE Symposium on Computers and Communications (ISCC)*, Jul. 2013, pp. 483–488.

[41] K. Thilina *et al.*, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2209–2221, Nov. 2013.

[42] K. W. Choi and E. Hossain, "Estimation of primary user parameters in cognitive radio systems via hidden Markov model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 782–795, Feb. 2013.

[43] T. Black, B. Kerans, and A. Kerans, "Implementation of hidden Markov model spectrum prediction algorithm," in *2012 International Symposium on Communications and Information Technologies (ISCIT)*, Oct. 2012, pp. 280–283.

[44] C.-H. Park *et al.*, "HMM based channel status predictor for cognitive radio," in *2007 Asia-Pacific Microwave Conference (APMC)*, Dec. 2007, pp. 1–4.

[45] Z. Chen *et al.*, "Channel state prediction in cognitive radio, part ii: Single-user prediction," in *2011 Proceedings of IEEE Southeastcon*, Mar. 2011, pp. 50–54.

[46] W. Yifei *et al.*, "QoS provisioning energy saving dynamic access policy for overlay cognitive radio networks with hidden Markov channels," *China Communications*, vol. 10, no. 12, pp. 92–101, Dec. 2013.

[47] A. Mukherjee, S. Maiti, and A. Datta, "Spectrum sensing for cognitive radio using blind source separation and hidden Markov model," in *2014 Fourth International Conference on Advanced Computing Communication Technologies (ACCT)*, Feb. 2014, pp. 409–414.

[48] A. Abdelmutalab, K. Assaleh, and M. El-Tarhuni, "Automatic modulation classification using polynomial classifiers," in *2014 IEEE 25th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2014, pp. 785–789.

[49] Y. Hassan, M. El-Tarhuni, and K. Assaleh, "Comparison of linear and polynomial classifiers for co-operative cognitive radio networks," in *2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2010, pp. 797–802.

[50] Y. Hassan, M. El-Tarhuni, and K. Assaleh, "Knowledge based cooperative spectrum sensing using polynomial classifiers in cognitive radio networks," in *2010 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec. 2010, pp. 1–6.

[51] Y. Hassan, M. El-Tarhuni, and K. Assaleh, "Learning-based spectrum sensing for cognitive radio systems," *Journal of Computer Networks and Communications*, vol. 2012, pp. 1–13, 2012.

[52] M. Gaafar, Y. Hassan, and T. Elshabrawy, "A hybrid cooperative spectrum sensing technique for cognitive radio networks using linear classifiers," in *2014 International Conference on Engineering and Technology (ICET)*, Apr. 2014, pp. 1–6.

[53] A. R. Webb, *Statistical pattern recognition*. New Jersey; West Sussex, England: Wiley, 2002.

[54] K. Narendra and K. Parthasarathy, "Learning automata approach to hierarchical multiobjective analysis," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 1, pp. 263–272, Jan. 1991.

[55] K. Murphy. (2001, Nov.) Hidden markov model (HMM) Matlab toolbox. [Online]. Available: http://nuweb.neu.edu/bbarbiellini/CBIO3580/HW7.html# usage [Accessed: Dec 27, 2013].

Vita

Menatalla Shehabeldin was born in 1992, in Ajman, in the United Arab Emirates (UAE). She received the B.Sc. degree in electrical engineering from the American University of Sharjah, UAE, with honors (magna cum laude), in 2013. She is currently pursuing her M.Sc. degree in electrical engineering at the American University of Sharjah, UAE.

Ms. Shehabeldin is currently a teaching assistant at the American University of Sharjah, UAE. Her research interests are wireless communications, next generation communications and cognitive radio.