Dynamic Spectrum Access Network Simulation and Classification of Secondary User Properties

Matthew John Rebholz

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Tamal Bose, Chair
Carl B. Dietrich
Majid Manteghi

19 August 2011
Blacksburg, Virginia

Keywords: Dynamic Spectrum Access, Cognitive Radios, Naïve Bayesian Classification, Matlab Simulation

DYNAMIC SPECTRUM ACCESS NETWORK SIMULATION AND CLASSIFICATION OF SECONDARY USER PROPERTIES

Matthew John Rebholz

ABSTRACT

This thesis explores the use of the Naïve Bayesian classifier as a method of determining high-level information about secondary users in a Dynamic Spectrum Access (DSA) network using a low complexity channel sensing method. With a growing number of users generating an increased demand for broadband access, determining an efficient method for utilizing the limited available broadband is a developing current and future issue. One possible solution is DSA, which we simulate using the Universal DSA Network Simulator (UDNS), created by our team at Virginia Tech.

However, DSA requires user devices to monitor large amounts of bandwidth, and the user devices are often limited in their acceptable size, weight, and power. This greatly limits the usable bandwidth when using complex channel sensing methods. Therefore, this thesis focuses on energy detection for channel sensing.

Constraining computing requirements by operating with limited spectrum sensing equipment allows for efficient use of limited broadband by user devices. The research on using the Naïve Bayesian classifier coupled with energy detection and the UDNS serves as a strong starting point for supplementary work in the area of radio classification.

This work was funded by the Rockwell Collins Advanced Technology Center of Cedar Rapids, IA.

Acknowledgements

I would like to acknowledge Benjamin Hilburn, who started work on this project and started the Universal Dynamic Spectrum Access Simulator (UDNS). I would like to thank Dr. Shubha Kadambe and the Rockwell Collins Advanced Technology Center of Cedar Rapids, IA, for their help with this research.

Finally, I would like to thank my committee members for their help in developing my thesis and working with me to defend on an unusual schedule. Especially I would like to thank my committee chair, Dr. Tamal Bose, who helped me through my graduate career.

Table of Contents

ABSTRACT i

Acknowledgements iii

Table of Contents iv

List of Figures vii

List of Tables ix

List of Abbreviations x

1 Introduction 1

1.1 US Radio Frequency Regulation 1
1.2 Unlicensed Frequency Bands 2
1.3 DSA as a Possible Solution 3

2 Background 4

2.1 Software Defined Radios 4
2.1.1 Cognitive Radios 5
2.2 Dynamic Spectrum Access 6
2.2.1 Channel Selection Criterion 7
2.2.2 Simplified Channel Selection Methods 8
2.2.3 Current Prediction Models 9
2.3 Naïve Bayesian Classification 10
2.3.1 Classification using Bayes Rule 10
2.3.2 Naïve Bayesian Classifier 12
2.3.3 Training for Continuous Inputs 13

3 Universal DSA Network Simulator Overview 15

3.1 Simulation Procedure 15
3.2 Timesteps 16
3.2.1 Visualization of Timesteps 17
3.3 Radio Definitions 18

3.3.1 Transmit/Idle Decisions 18
3.3.2 Channel Selection Method 19
3.3.3 Other Properties 19
3.4 Channel Simulation 20
3.4.1 Noise Generation 20
3.4.2 Symbol Generation 20
3.4.3 Upsampling 21
3.4.4 RRC Filtering 22
3.4.5 Path Loss 26
3.5 Signal Detection 27
3.5.1 Cabric's Energy Detector 27
3.5.2 Energy Detector Use 31
3.5.3 Implementation of the Energy Detector 31
3.5.4 Sub-timestep Signal Detection 32
3.6 Data Recording 34

4 Experimentation Profile 38

4.1 Standard Simulation Profile 38
4.2 Visualization of a Simulation Run 41
4.2.1 Primary Users 42
4.2.2 Secondary Users 43
4.2.3 Channel Accesses per Secondary User 44
4.2.4 SU Channel Selections 45

5 Classification of Secondary User ID 48

5.1 Motivation 48
5.2 Classification Objectives 48
5.2.1 The Classifier: NaiveBayes 49
5.3 Classification Method: Direct Classification 50
5.3.1 Train/Test Length 51
5.3.2 Direct Classification Results 53
5.3.3 Energy Detector Deactivation 53
5.3.4 ED Deactivation Results 54
5.4 Classification Method: Hard-decision Pre-processing 55
5.4.1 Hard Decision Testing 55
5.4.2 Decision Mapping 57
5.5 Classification Method: Negative Flagging 59
5.5.1 Negative Flagging Results 60

5.5.2 Increasing the Number of SUs 61
5.6 Accuracy and Channel Utilization 63

6 Classification of Channel Selection Method 65

6.1 Classification Objectives 65
6.2 Classification Method 65
6.2.1 Multiple SUs using the Same Channel Selection Method 66
6.2.2 Backward Classification 68
6.3 SU Grouping: Classifier Histogram 68
6.3.1 Histogram Setup 69
6.3.2 Creating the Similarity Matrix 70
6.3.3 Removing False Groupings 71
6.4 SU Grouping: Bayesian Probability 73
6.4.1 Classification Method 74
6.4.2 Detection Threshold Calculation 75
6.4.3 Two SUs Per Selection Method Results 80
6.4.4 Different Numbers of SUs Per Selection Method Results 82

7 Conclusion 87

7.1 Future Work 88

References 89

List of Figures

Figure 1: Simplified block diagram comparing an SDR receiver to a traditional radio receiver. . 5
Figure 2: Visualization of timestep organization of radios over time. 17
Figure 3: QPSK constellation diagram. 21
Figure 4: RRC filter impulse response. 23
Figure 5: Frequency response of the RRC filter 24
Figure 6: FFT of the filtered message without noise. 25
Figure 7: FFT of the filtered message with noise. 26
Figure 8: Histogram of the energy detector with 16 input samples. 28
Figure 9: Histogram of the energy detector test statistics with 256 input samples. 29
Figure 10: Radio operating curves for 256 samples 30
Figure 11: Sub-timestep probability of signal detection. 33
Figure 12: Histogram of ED test statistics for sub-timestep detection – 0 dB SNR. 34
Figure 13: Channel occupancy rate for PUs only. 42
Figure 14: Channel occupancy rates for all radios. 43
Figure 15: SU interference rates by channel number. 44
Figure 16: Channel access count by user. 45
Figure 17: Count of channel selections by channel selection method. 46
Figure 18: Channel evacuation properties by channel selection method. 47
Figure 19: Visualization of the direct classification method. 51
Figure 20: Estimation accuracy against ratio of training to test lengths. 52
Figure 21: Bayesian classifier accuracy using the direct classification for three and four SUs in a 10 PU network. 53
Figure 22: Bayesian classifier accuracy using hard-decisions for three and four SUs in a 10 PU network. 57
Figure 23: Pre-processed decision metric mapping. 58
Figure 24: Classifier accuracy using nine hard-decision, pre-processed points. 59
Figure 25: Classifier accuracy using the negative flagging method for three and four SUs in a 10 PU network. 61
Figure 26: Classification accuracy of four and eight SUs in a 10 PU network. 63

Figure 27: Classification accuracy as a function of spectrum utilization. 64

Figure 28: Classification accuracy for four and eight SUs when determining accuracy of estimating the proper channel selection method. 67

Figure 29: Histogram of SU ID estimations, plotting the number of estimated SU IDs for each truth SU ID over 5,000 channel decisions. 69

Figure 30: Histogram of SU ID estimations, plotting the number of truth SU IDs for each estimated SU ID over 5,000 channel decisions. 70

Figure 31: Normalized difference between the SUs over 5,000 channel decisions. 71

Figure 32: Normalized difference between the SUs with discarded inputs over 5,000 channel decisions. 72

Figure 33: Normalized difference between the SUs as calculated by the classifier with discarded inputs over 200 channel decisions. 73

Figure 34: Classifier probability outputs for SU 13 and 14 for channel decisions made by SU 13 as a function of time. 75

Figure 35: Probability of missed group detection for a range of threshold values using a single classifier output from an SU 13 channel selection. 77

Figure 36: Probability of missed group detection for SU 14 for a range of threshold when averaging over multiple channel selections by SU 13. 78

Figure 37: Probability of a false group detection. 79

Figure 38: Simulation results for group detection. 81

Figure 39: Channel group detection for two SUs per channel selection method. 82

Figure 40: Group detection for more than two SUs per channel selection method. 84

Figure 41: Probability of missed detection for the new simulation case. 85

Figure 42: Group detection for more than two SUs per channel selection method with a lowered threshold value. 85

List of Tables

Table 1: Simulation Outputs 35
Table 2: List of SU Status Codes 36
Table 3: Radio Statistics Recorded 37
Table 4: Universal Simulation Values, Sample Rate and Timestep Duration in Samples 39
Table 5: Standard Simulation PU Properties 39
Table 6: Standard Simulation SU Properties, with * Marking Values that were Modified During Classification Testing 40
Table 7: SU Channel Selection Methods for Three and Four SU Simulations 41
Table 8: Standard Simulation Channel Noise Powers 41
Table 9: Distributions Built in to the Matlab NaiveBayes Function 49
Table 10: Eight SU Simulation Properties 62
Table 11: Statistics for First SU Grouping Test 82
Table 12: SU Properties for Final Simulation Testing 83
Table 13: Statistics for Second SU Grouping Test 84
Table 14: Statistics for Second SU Grouping Test with Lowered Threshold Value 86

List of Abbreviations

ADC:	Analog to digital converter
AR:	Autoregressive
AWGN:	Additive white Gaussian noise
CR:	Cognitive radio
CRN:	Cognitive radio network
DAC:	Digital to analog converter
DCS:	Dynamic channel selection
DSA:	Dynamic spectrum access
ED:	Energy detector
FCC:	Federal Communications Commission
GPS:	Global positioning system
ID:	Identification
ISI:	Inter-symbol interference
LABT:	Listen and avoid before talk
M-PSK:	M-ary phase-shift keying
OFDM:	Orthogonal frequency division multiplexing
PDF:	Probability density function
PSK:	Phase-shift keying
PU:	Primary user
QoS:	Quality of service
QPSK:	Quadrature phase-shift keying
SNR:	Signal-to-noise ratio
SOM:	Self-organizing map
SU:	Secondary user
UDNS:	Universal DSA Network Simulator
US:	Unites States
VT:	Virginia Tech
WLAN:	Wireless local-area network

1

Introduction

This section covers the background information on the regulation of radio frequencies in the United States (US).

1.1 US Radio Frequency Regulation

The US congress created the Federal Communications Commission (FCC) in 1934, and was tasked with, among other things, the management and licensing of electromagnetic spectrum. Since then, most of the US spectrum, such as the frequencies that are used by broadcast television, has been sold to licensed users.. These licenses are sold to specific users for a fixed location and region, with the FCC enforcing the policies for these bands.

However, in the past 10 years, there has been an explosion in the number of personal wireless devices. These devices range from mobile phones to wireless internet routers. While some of these devices use FCC-licensed bands, such as global positioning systems (GPS) systems, other systems such as Bluetooth and Wi-Fi use unlicensed bands.

The FCCs general policy has not changed dramatically since its creation. This policy is to provide important services, and likewise protect those services from harmful interference, which has resulted in long-term exclusive use of spectrum over large areas. One

example is over-the-air (OTA) television (TV) The inclusion of political inefficiencies further hampers spectrum utilization [2].

1.2 Unlicensed Frequency Bands

The addition of mobile, personal wireless devices makes regulation difficult. Individually licensing each individual device within a Wi-Fi wireless local area network (WLANs) in a person's home would be challenging and expensive. Therefore, these systems have been forced largely to operate using unlicensed frequency bands, as well as abide by output power and other constraints, set by the FCC.

The most popular unlicensed bands are the industrial, scientific, and medical (ISM) bands. In the US, these bands are located at 900 MHz, 2.45 GHz, and 5.8 GHz. They are controlled by Part 18 of the FCC rules, with the rules pertaining to the unlicensed use governed by Part 15. Part 15 states that unlicensed, spread spectrum users are permitted to use the ISM bands listed above if they operate on a non-interference basis and the maximum output power is limited to 1 Watt [3].

With more people using these devices, especially in populated areas, the competition for bandwidth causes problems. WLANs, cordless phones, and Bluetooth devices all provide interference to one another, with additional interference from sources such as microwave ovens.

However, the crowding of the unlicensed bands is in stark comparison to a majority of the licensed spectrum in the US. Two studies conducted by the FCC showed a considerable amount of white space, or unused spectrum, is available between 30 MHz and 3 GHz. The

study also found that, of the frequency bands that were in use, that between 15 percent and 85 percent of their potential was being utilized. The latter figure being highly dependent on the location of the receiver and the time of day in which the measurements are taken [4].

1.3 DSA as a Possible Solution

Some examples of these underutilized channels are the bands associated with public safety, military, and government, as well as certain TV bands. Some of these bands are highly utilized, but this utilization depends heavily on the geographic location [5]. One possible solution to make better use of these bands is the implementation of DSA, which would allow unlicensed users to take advantage of the unused spectrum

This thesis explores the Universal DSA Network Simulator (UDNS) created by our team at Wireless@VT and its application to simulating the use of DSA in real-world systems. Because of the potential interference caused by unlicensed users, this thesis will explore the usefulness of simple, low-cost channel measurement techniques for the classification of other radios.

To be able to adapt to the changing channel requirements, the radios need to be very flexible. One potential class of radio to fulfill this is discussed in the Chapter 2.

2

Background

This chapter covers the background of software defined radios (SDR) and provides a review of the channel selection methods used by cognitive radio networks. The Naïve Bayesian classification process, which is tested in later chapters for estimating secondary user properties, is also explained.

2.1 Software Defined Radios

The idea of software defined radios was first introduced by Mitola in 1993 [6]. The basic design of an SDR receiver usually starts with a radio-frequency (RF) front-end with a software-controlled tuner. The RF front-end mixes the signal down to baseband or a lower intermediate frequency, and applies the necessary filtering. Once at the lower frequency, the signal is passed into an analog-to-digital converter, which converts the continuous signal into quantized values. This sampled signal can then be handed off to a general device, such as a general purpose processor (GPP), or a more specialized device such as a field-programmable gate array (FPGA) or digital signal processor (DSP).

A block diagram of an SDR receiver can be found in Figure 1 alongside a simplified block diagram for a traditional radio. This shows, at a high level, the steps that are accomplished in software rather than in hardware. This allows the radio to be configured

without replacing hardware components, which is considerable cheaper. Not pictured is the frequency control going from software to hardware.

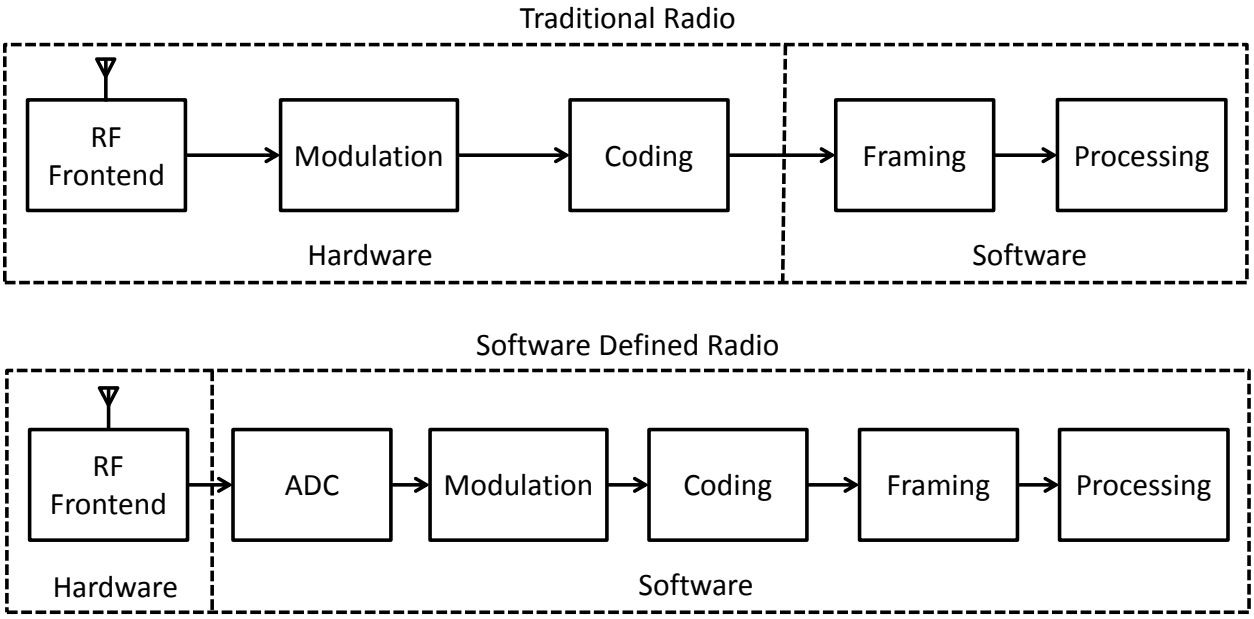

Figure 1: Simplified block diagram comparing an SDR receiver to a traditional radio receiver.

The added flexibility of an SDR can be advantageous for decreasing the lead time needed to design radio systems. Another possibility is that the SDR could automatically configure itself to meet or could be used to modify the properties of the radios in the field to meet the needs of their conditions, which requires a new class of SDR.

2.1.1 Cognitive Radios

In 1999, Mitola defined the phrase cognitive radios (CR) [9]. A CR is a radio that has both the ability to sense its environment and the capability to react to changes or specific situations. While in recent history, the term cognitive radio has become popular for a multitude of SDR functionalities, Mitola's research was directed at high-level user functionality. One example is adjusting the quality of service (QoS) guarantees to match the

urgency of a call, which could be estimated by taking advantage of the users previous actions [10].

CRs are the subject of intensive research due to the potential of increasing the throughput of future wireless communications. The ability for a CR to sense its environment and act upon that data allows the individual radios, or networks as a whole, to take advantage of white space and sparsely used channels. In addition to the FCC findings noted in Chapter 1, a test of spectrum utilization during a political convention was conducted in New York City from August 31 to September 1, 2004. The test was conducted during an expected time of higher than usual wireless activity, in a major city, and showed that only about 13 percent of the spectrum was being utilized [2]. Such low utilization could be improved by CRs, which could take advantage of the white space.

To take advantage of the unused spectrum, CRs are combined to form cognitive radio networks (CRN). In a general sense, these networks are populated by two types of users: licensed users, referred to as primary users (PU), and unlicensed users, referred to as secondary users (SU). The primary users own the license to use the particular channel or spectrum band, and the addition of the SUs to the network should not interfere with the actions of the PUs [11].

2.2 Dynamic Spectrum Access

There are two main classes of channel selection models for the SUs within a CRN. These two classes are dynamic channel selection (DCS) and dynamic spectrum access (DSA). DCS differs from DSA in that the former uses an arbiter to allocate the channels to SUs inside the network [12], whereas in a DSA system, the secondary users (SU) are responsible for

selecting their own channels. This is a very high-level description of a DSA system, but is true for most implementations.

Even within DSA, each network can have very different methods of implementation. One difference is the method of spectrum sensing. The networks discussed in [13] and [14] use a distributed spectrum sensing approach, where each SU reports back to the base station to determine the availability of a channel using simple energy detectors, whereas [15] discusses the use of cyclostationary detection without the use of multiple testing nodes.

The signals transmitted can also vary greatly. The model used in [16] includes additional complexities in using cyclostationary signatures in multi-carrier signals in order for the receive node to be able to establish a communication link, without requiring a separate control channel. The contrasts to the model discussed in [17], which relies on a control channel.

The application can also vary considerably. Many papers focus on personal wireless devices, and while others such as [18] and [19] focus on inter-vehicle communications. These all show the lack of a concrete definition, which would normally accompany a radio system, such as the protocols associated with Institute of Electrical and Electronics Engineers (IEEE) 802.11 standard. This is because a final functionality has not yet been determined.

2.2.1 Channel Selection Criterion

This section will survey some of the channel selection techniques that have been proposed for DSA networks, as well as DCS networks. Channel selection methods for DCS networks were included to broaden the scope of the survey. In this section, papers with similar channel selection criteria are grouped together. For a large number of the papers, the

selection of the channel selection criteria depends on the specific situation that the paper was seeking to address.

The authors of [20] suggest that channels be described by parameters of interference, path loss, link layer delay, and PU activity. These are good overall parameters for analyzing a radio network. Using more narrow channel criteria, [12] suggests signal strength and collision probability as primary parameters. In a similar narrow approach, [19] suggests maximum bit rate and physical location of the nodes.

The authors of [21] focus on the use of orthogonal frequency division multiplexing (OFDM) with a listen and avoid before talk (LABT) method to transmit the SU's message in the white space left by unoccupied, but not necessarily adjacent, channels. However, [22] discusses the use of a cooperative method of channel selection using a Bayesian classifier to achieve cooperation between the SUs, and decrease the amount of time needed to transmit a given message for all of the radios.

The final method to be discussed is [23]. The authors of [23] use a Neural Network to better understand the network conditions and their effect on performance in each channel. From this data, a channel selection can be made.

2.2.2 Simplified Channel Selection Methods

The channel selection techniques chosen to be implemented in the UDNS represent simple building blocks of the parameters mentioned above. The choice to do this was two-fold. The first reason is to determine a baseline to which later tests with more complex criteria and channel selection algorithms can be compared. The second is to test the ability of the classification with less-complex inputs. This is a useful test in determining estimation

accuracies for reduced-complexity cases, and therefore the feasibility of accurate channel estimation given a minimal spectrum sensing capabilities.

The channel selection methods chosen were least energy, least frequently occupied, least recently occupied, and least accessed. For this thesis, a channel is occupied whenever a radio is currently transmitting on it; however, channel access is defined as the act of a radio beginning to transmit a message after being idle. A radio first accesses a channel, then occupies it, which it can do for multiple transmit blocks.

These channel selection strategies were chosen because they are included in some form in all of the methods mentioned in the previous section. Therefore, these can be thought of as building-blocks, and used to create more complex methods later in the development of the simulator. Negatively, these methods provide no means of spectrum sharing, meaning that there is currently no mechanism for the SUs to stop one SU from dominating the network utilization time. This results in a degradation of the overall throughput, as described in [3].

2.2.3 Current Prediction Models

This section will briefly cover the current field of prediction and simulation models that exist for DSA networks. Of the papers found on prediction within a DSA network environment, most were based on the prediction of the mobility of the other radios. In [25] the authors attempt to predict the time of the next action by a CR, either by channel selection or ending its transmission using either a Markov family predictor, a moving-average predictor, a CDF predictor, or a static neighbor graph predictor.

Their findings show a slight increase in the successful transmission of packets using a prediction model. The authors of [26] attempted to do the same for an IEEE 802.11 system

using an autoregressive (AR) model, and saw positive results due to the system having a time series with characteristics of an AR model.

2.3 Naïve Bayesian Classification

For the classification portion of this thesis, the Naïve Bayesian classifier is used. This method of classification was chosen because of its ability to quickly process a large number of inputs, as well as its low cost of operation on an end-user mobile. The Naïve Bayesian classifier is known as an effective method for classifying documents, such as spam filtering [27], and has shown positive results in gene expression patterns [28].

2.3.1 Classification using Bayes Rule

Bayes' rule is a theorem named for Thomas Bayes that allows one to calculate the inverse conditional probability by

$$\boldsymbol{P(A|B) = \frac{P(B|A)P(A)}{P(B)}} \quad \textbf{(1)}$$

which can be expanded to

$$\boldsymbol{P(A_i|B_k) = \frac{P(B_k|A_i)P(A_i)}{\sum_j P(B_k|A_j)P(A_j)}} \quad \textbf{(2)}$$

where A_i represents the i^{th} value of A within the A vector, B_k represents the k^{th} value of B within the B vector, and $\sum_j$ is the summation over all possible values of A. These are useful when calculating the value of P(A|B) in situations where that value cannot easily be calculated directly, but one has the ability to easily calculate P(B|A). This section will describe the use of this theorem in the design of a learning algorithm.

Initially, we will assume that Y is a Boolean random variable, and $X = < X_1, X_2, X_3, \ldots, X_N >$, where X_i is a Boolean random variable, with i denoting the i^{th} attribute of X. Now we will approximate the target function P(X|Y). Applying equation (2), the probability of y_i given the k^{th} value of X within the X vector can be calculated as

$$\boldsymbol{P(Y = y_i | X = x_k) = \frac{P(X = x_k | Y = y_i) P(Y = y_i)}{\sum_j \left(P(X = x_k | Y = y_j) P(Y = y_j) \right)}} \quad \textbf{(3)}$$

where capital letters represent the random variables, lower case letters represent the value of the random variable, and the summation in the denominator is taken over all possible values of Y. This can be used to learn P(Y|X) by using training data to estimate P(X|Y) and P(Y).

However, the amount of training data needed to reliably estimate the distributions must first be determined. For this, assume we wish to estimate

$$\boldsymbol{\theta_{ij} = P(X = x_i | Y = y_j)} \quad \textbf{(4)}$$

where i is the index of the possible values of X, and j is the index of the possible values of Y. Assuming that X is a Boolean vector with a length of N, i will have 2^n possible values, and j will have two possible values for the two possible states of Y. This means that a total of 2^{N+1} parameters need to be estimated. However, since for any fixed j, the summation of θ must equal one, only $2(2^N-1)$ estimates need to be made, with the final value being calculated by the difference between the sum of the other values and one. The leading value of two represents the two possible values of Y.

Assuming an X vector length of 40, this would require approximately $2.2*10^{12}$ parameters to be estimated. Additionally, to obtain reliable estimates of each of these parameters, multiple instances must be observed. This further increases the number of

observations which are needed, and becomes too large for the large-dimension inputs in realistic systems.

2.3.2 Naïve Bayesian Classifier

To reduce the number of parameters that need to be estimated, the Naïve Bayesian classifier makes a conditional independence assumption. This assumption states that, given random variables X, Y, and Z, X is conditionally independent of Y given Z. This is the case if and only if the probability density function (PDF) governing X is independent of Y.

Using this assumption drastically decreases the number of parameters which need to be calculated. Rather than the number of parameters increasing exponentially following the equation $2(2^N-1)$, the number of parameters only increases linearly following the equation 2N. This is a clear advantage for situations when a large number of inputs are required.

Using this conditional independence assumption, we can determine that when X has N attributes that are conditionally independent given Y, the resulting equation is

$$\boldsymbol{P(X_1 \dots X_N|Y) = \prod_{i=1}^{N} P(X_i|Y)} \quad \textbf{(5)}$$

and substituting equation (5) into equation (3), it can be rewritten as

$$\boldsymbol{P(Y = y_k|X_1 \dots X_N) = \frac{P(Y = y_k) \prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i|Y = y_j)}} \quad \textbf{(6)}$$

which is the fundamental equation for the Naïve Bayesian classifier. This equation shows how the probability Y will take on a given value for a set of observed X values, given the distributions of P(Y) and $P(X_i|Y)$.

If we are only interested in the most probable value of Y, then equation (6) can be modified to

$$Y \leftarrow \boldsymbol{argmax}_{y_k} \frac{\boldsymbol{P(Y = y_k)} \prod_i \boldsymbol{P(X_i | Y = y_k)}}{\sum_j \boldsymbol{P(Y = y_j)} \prod_i \boldsymbol{P(X_i | Y = y_j)}} \quad \textbf{(7)}$$

which can be further simplified to

$$Y \leftarrow \boldsymbol{argmax}_{y_k} \; \boldsymbol{P(Y = y_k)} \prod_i \boldsymbol{P(X_i | Y = y_k)} \quad \textbf{(8)}$$

because the denominator of equation (7) does not depend on y_k.

2.3.3 Training for Continuous Inputs

When the inputs to the classifier are continuous, equations (6) and (8) can still be used as the basis for designing a Naïve Bayesian classifier. However, a different method of representing the distributions is needed. A common method is to assume that the distribution of each continuous X_i is Gaussian and can be defined by a mean and standard deviation.

This requirement does slightly increase the number of parameters that need to be estimated, resulting in a total of 2NK parameters, where N represents the number of X attributes, and K represents the number of discrete states within Y. The leading two in the parameter calculation accounts for the mean and standard deviations being calculated separately.

To calculate the maximum likelihood estimates, the maximum likelihood estimator for the mean is

$$\hat{\boldsymbol{\mu}}_{ik} = \frac{\mathbf{1}}{\sum_j \boldsymbol{\delta}(\boldsymbol{Y^j = y_k})} \sum_j \boldsymbol{X_i^j \delta(Y^j = y_k)} \quad \textbf{(9)}$$

where $\hat{\mu}_{ik}$ is the likelihood estimate for μ_{ik}, i represents the X vector index, j represents the j^{th} training sample, and $\delta(Y = y_k)$ is one if $Y = y_k$ and is zero otherwise. Following a similar principle, the maximum likelihood estimator for the variance is

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j \left(X_i^j - \hat{\mu}_{jk}\right)^2 \delta(Y^j = y_k) \quad \textbf{(10)}$$

where $\hat{\sigma}_{ik}^2$ is the maximum likelihood estimate of the variances of X for each Y. The rest of the terms are the same as equation (9).

3

Universal DSA Network Simulator Overview

This chapter provides an in-depth review of the Universal DSA Network Simulator (UDNS) that was developed to accurately simulate a wide variety of network conditions, using the network properties described in the previous chapter for the simplified environment. Each of the following sections provides information about the functions and operations of the simulator, which were originally introduced by Benjamin Hilburn in [29].

3.1 Simulation Procedure

The UDNS is a Matlab simulation environment. It is initiated from an M-file front-end that is a temporary GUI until a proper GUI can be built. This M-file contains all of the default values and options that control the actions of the simulator, which are all available to the user for modification. When the M-file is be run, user settings are stored, threshold values for the energy detector are calculated, and it starts the simulation function.

When the simulation function is first called, the SUs are idle for a period of time while the PUs are allowed to operate freely on their given channels. This initial phase allows the PUs to operate without the presence of the SUs to achieve a steady state. This approximates the environmental conditions that new SUs would encounter if they were to be activated in a new environment.

Once the SUs are activated, the PUs operate as if the SUs were not present in the channels, with the SUs evacuating the channel once that channel's PU begins to transmit. When the SU attempts to transmit again, the SU will determine which channels are unoccupied by any radios, analyze the channel statistics for all of the unoccupied channels, and make a channel selection decision based on that SUs channel selection strategy.

The length of time that the UDNS operates is governed by one of two options. The first option is the length of time, which is measured in units of timesteps that should be simulated. The second is the number of channel selection decisions that the simulator should observe, which is the number of times a SU makes a decision of which channel to use. The simulator can be told to run until either one of these has been reached, or both have been reached.

When the simulation is complete, the simulator returns all useful data for analyzing the performance of the network, actions of the radios, and all of the final states of the simulator. By saving the final states of the simulator, this allows the user to either continue the same simulation at a later date, or to use some of the conditions from a previous run to kick-start another run.

3.2 Timesteps

All of the transmissions in the UDNS are based on the simplifying assumption that all transmissions begin and end at discreet intervals, as observed by a hypothetical SU that does not transmit. Therefore, the simulations run in the UDNS do not run in continuous time, but rather in units of timesteps, which represent a fixed time duration depending on the number of samples simulated during each timestep and the rate at which the timesteps are observed.

To allow for greater flexibility in the simulator, the number of samples observed per timestep and the sample rate are user-defined at run-time. The amount of time, in seconds, that an individual timestep represents depends on the sample rate of the hypothetical SU. The amount of time observed during each timestep can be calculated by multiplying the number of samples per timestep by the sample rate in samples per second.

3.2.1 Visualization of Timesteps

Figure 2 shows a visualization of the presence of the radios in the simulation environment within the timestep framework. Each column represents a different channel and each row represents a timestep. Therefore, each square represents the condition of that channel during that timestep, where a radio ID number is listed in the space to represent the presence of that radio. Additionally, each space is colored in accordance to the type of radio occupying the channel. Green represents a PU with a radio ID number of 10 or less, while yellow represents a SU with a radio ID number of 11 or more.

		Channel Number									
		1	2	3	4	5	6	7	8	9	10
Timestep	N		2	3	16	5		13	8	11	10
	N+1		2	3	16	5		13	8	11	10
	N+2			3	16	5		13	8	11	10
	N+3			3	16	5		13	8	11	10
	N+4			3	16	5		13	8	11	
	N+5			3	16	5	6	13	8	11	
	N+6			3	16	5	6	13	8	11	
	N+7	14		3	16	5	6	13	8	11	
	N+8	14		3	16	5	6	7	8	11	
	N+9	14		3	16	5	6	7	8	11	
	N+10	14		3	16	5	6	7	8	11	

Figure 2: Visualization of timestep organization of radios over time.

3.3 Radio Definitions

Before the simulation begins, the user can set the properties of each radio, each of which is described in the following sub-sections. For all radios, these properties include if the radio is a PU or SU, the SNR of the radio, the transmit lambda, the idle lambda, the bit rate, the alpha value for root-raised cosine (RRC) filtering, and the number of bits per symbol. PUs have the additional property of which channel that radio is licensed to use. SUs have the additional properties of what channel selection method they will use, a Poisson mean that is used to determine when a SU will change channel selection methods, what channels the SU will monitor, and what channel selection strategies the SU can choose from.

3.3.1 Transmit/Idle Decisions

As described in the previous section, the simulation is conducted in units of timesteps. The transmit decisions are made according to two different counters, a transmit and an idle counter. When a radio is idle, that radio's idle counter is decremented for every timestep the radio is idle. Once the idle counter reaches zero, the radio will either begin transmission in the case of a PU, or look for an available channel in the case of a SU. If an SU cannot find an available channel, the radio will reset its idle counter and try again.

Once a radio has begun transmitting, the activity lambda is used to generate the number of timesteps the radio will transmit. While the radio transmits, the transmit counter is decremented for every timestep the radio transmits until the transmit counter reaches zero. Once a radio has begun to idle, the idle lambda is used to generate the number of timesteps the radio will transmit.

3.3.2 Channel Selection Method

SUs also contain a channel selection method counter. Each time the SU makes a channel decision, the channel selection method counter is decremented. When the counter reaches zero, the SU will choose a different channel selection method from its list of possible channel selection methods, and the counter is reset using a Poisson distribution with a user-defined mean.

3.3.3 Other Properties

The last properties set for the users are those which control the transmitted signal. These are the signal-to-noise-ratio (SNR), number of bits per symbol, bit rate, and alpha value. The signal power of the SUs change depending on which channel the user occupies, giving each SU a fixed SNR that is independent of the current channel, since each have a different noise floor. The signal powers of the PUs do not change because of the fixed noise power of their assigned channel and their fixed SNR. The modulation used for all transmissions is an M-PSK scheme, where M is controlled by the equation

$$\boldsymbol{M = 2^b} \tag{11}$$

where *b* is the number of bits per symbol, and is set by the user in the radio definitions.

At the end of a simulation run, the current state of all of the radios is returned to the main simulation control file. This allows the current state of the radios to be reused in a future simulation. This can be used to either prime a second simulation using a previous steady-state to determine variations in simulations given the same initial radio conditions, or to continue the same simulation at a later time, which is described in the following section.

3.4 Channel Simulation

The UDNS determines the number of channels to be simulated from the number of PUs defined for the simulation. This value can be any arbitrary number, and is due to the property of DSA networks that each PU has its own channel. If a simulation is required that includes channels with no assigned PU, this can be simulated by defining a PU for that channel that has an activity lambda of zero. This signals the UDNS to create a channel with an inactive PU.

The only feature not described in the radio definitions for the channels is the noise power of each channel. This is set by the user in dBm, and is static throughout the simulation.

3.4.1 Noise Generation

When a channel is simulated, this noise power is used to generate the AWGN for each channel. The noise samples are generated as two independent vectors of the in-phase and quadrature samples with zero mean.

3.4.2 Symbol Generation

The current version of the UDNS does not have features to allow for the reception of data for calculating error rates. Therefore, all data that is generated is random and used only to simulate the channel. This is generated only if the channel being simulated is currently occupied by a radio. This is done using a three-step method. The first step is generates a stream of random data. The data matrix has the same number of rows as the number of bits per symbol and the same number of columns as the number of symbols to be transmitted during that timestep.

Once a random data stream has been generated, the second step maps the binary values into symbols indices. This generates a vector symbol from the matrix of bits. The symbols are then mapped to angles in radians and converted into complex coordinates along the edge of the unit circle. Figure 3 shows the constellation diagram for QPSK, plotted in I-Q space. The current version of the UDNS does not apply Gray coding to the symbols since the data is only used to simulate the transmitted messages. Later versions of the simulator may add M-QAM to the simulation.

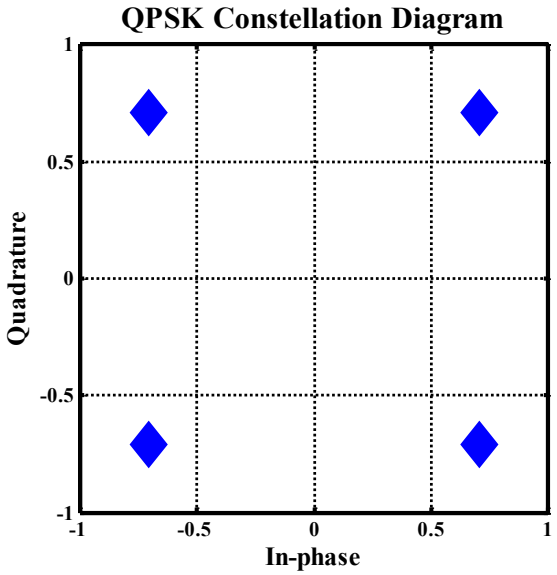

Figure 3: QPSK constellation diagram.

3.4.3 Upsampling

At this point, the data has been generated and manipulated at one sample per symbol. The next step is to upsample the data to achieve the desired sample rate. The current version of the UDNS only supports systems where the sample rate is an integer multiple of the symbol rate. Future versions may include sub-integer multiples, but for the general simulations conducted without data tracking, integer multiples provide a good baseline.

Upsampling is achieved by inserting L-1 zeros between each symbol, where L is the upsampling factor, which is determined by the ratio between the desired sample rate and the symbol rate. The samples going into the upsampler have unit variance, so to maintain this property, which is required later in the simulation, the upsampler multiplies the upsampled message by L. At this point, the upsampled signal has L aliases of the message in frequency. The UDNS uses the pulse-shaping filter to remove these aliases.

3.4.4 RRC Filtering

To maintain strict band-limiting so the SUs do not interfere with radios using adjacent channels, a strict band-limited pulse shape was chosen. For this, the raised-cosine is the ideal pulse shape, since it is strictly band-limited and introduces no inter-symbol interference (ISI). However, when matched filtering is applied at the receiver, the convolution of the two raised-cosines loses introduces ISI. To correct for this, the RRC waveform is used.

Using the property that convolution in the time domain is multiplication in the frequency domain, the RRC pulse can be determined by taking the square root of the frequency response of a raised-cosine filter. When matched filtering is applied at the receiver, the convolution of the two RRC pulses results in a raised-cosine. This filtered signal has zero ISI, even though the transmitted RRC message has ISI.

At this time, the UDNS only incorporates RRC pulse shaping. To generate the RRC pulse, the Matlab function ‘firrcos’(n,Fc,alpha,Fs,'rolloff','sqrt’) is used. This function is built-in to Matlab and generates an RRC pulse for a given filter order (n), cutoff frequency (Fc), rolloff factor (alpha), and sample rate (Fs). The tag ‘rolloff’ at the end is indicates that a rolloff factor is specified rather than the transition bandwidth and ‘sqrt’ indicates that an RRC

filter should be designed rather than a raised-cosine. The filter is applied to the baseband message.

The order of the RRC filter used in the UDNS is determined by the equation

$$F_{order} = \frac{5R_s M}{R_b}$$
$$F_{order} = F_{order} + mod(F_{order}, 2) \quad (12)$$

where F_{order} is the filter order, R_s is the sample rate, M is the number of bits per symbol, R_b is the bit rate, and mod is the modulus function. The modulus function is used to enforce the fact that the order of the RRC filter is even. This fixes the length of the filter to be five symbol periods long.

Figure 4 shows the impulse response of the default RRC filter used in the UDNS, and the ISI introduced at samples 0, 5, 10, 15, and 20. This filter has a filter order of 20, a rolloff factor of 0.5, and a cutoff frequency equal to the symbol rate, which is one quarter of the sampling rate in example. The filter order and the cutoff frequency change depending on the number of bits per symbol, the symbol rate, and the sample rate.

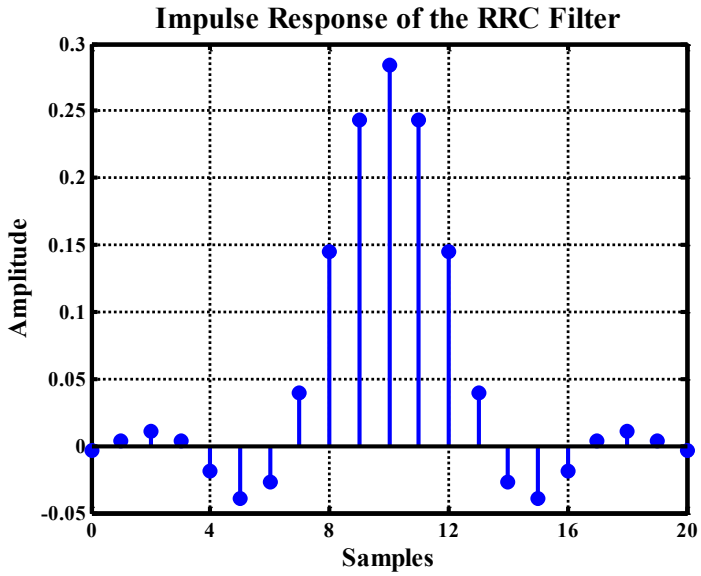

Figure 4: RRC filter impulse response.

Figure 5 shows the frequency response of the RRC filter pictured in Figure 4. This particular filter shows side band suppression of at least 15 dB, which is acceptable in the simulation assuming that the simulation will be run using SNR values less than 15 dB. If a higher signal power or lower noise floor is needed, the order of the RRC filter needs to be increased.

Figure 5 also shows the effect of the 0.5 rolloff factor, which controls the rolloff of the frequency response outside of a bandwidth equal to the symbol rate, which is one quarter of the sample rate. This corresponds to a fractional bandwidth of 0.25. Decreasing the rolloff reduces the amount of additional bandwidth occupied by the signal, but at the cost of requiring a longer filter to suppress side bands.

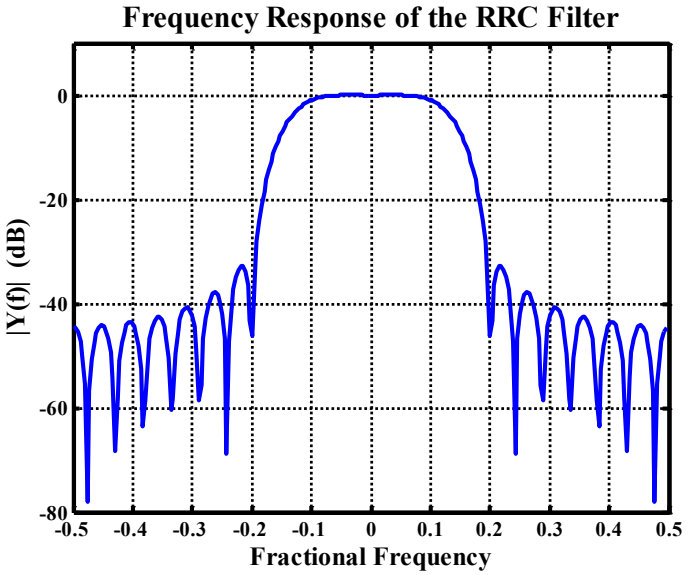

Figure 5: Frequency response of the RRC filter

Once the filter has been calculated, the output of the upsampler is filtered using the RRC filter. This will suppress the images by at least 15 dB for this RRC filter design. The resulting signal has the spectral qualities shown in Figure 6, which was generated using a Monte Carlo simulation with 200 independent runs, and each run used 1,000 QPSK symbols.

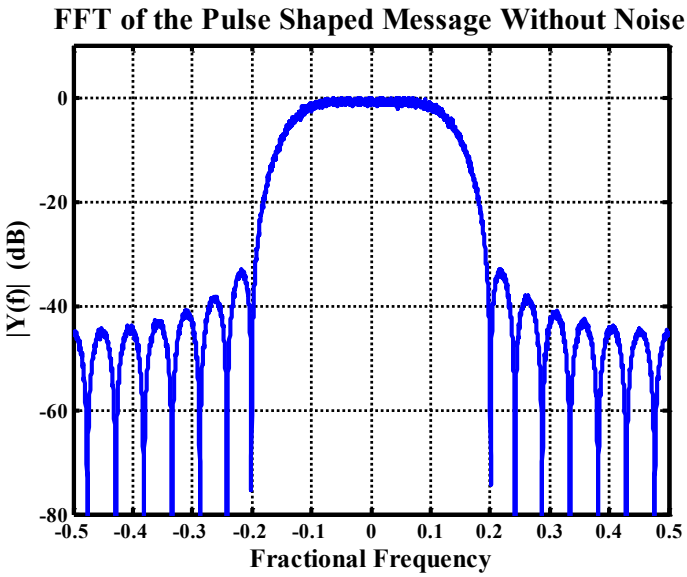

Figure 6: FFT of the filtered message without noise.

Once the signal has been filtered, the signal is set to the proper amplitude. Since the simulated data is sent using M-PSK modulation, the information is only encoded in the angle of the symbol, which allows the amplitude to remain constant. The symbol power is set to unity in the upsampling stage. This allows the signal power to be set using the equation

$$\boldsymbol{y(n) = 10^{\frac{SNR+\sigma_N^2}{20}} * x(n)} \tag{13}$$

where x(n) is the output of the upsampler, SNR is the SNR of the radio, σ_N^2 is the noise power in dBm, and y(n) is the transmitted message with the proper amplitude.

Figure 7 shows the signal spectrum with noise present. The signal shown has an SNR of 5 dB, noise power of 0 dBm, bit rate of 500 kHz, sample rate of 1 MHz, two bits per symbol, and a rolloff factor of 0.5. Figure 7 was generated using a Monte Carlo method with 200 independent runs.

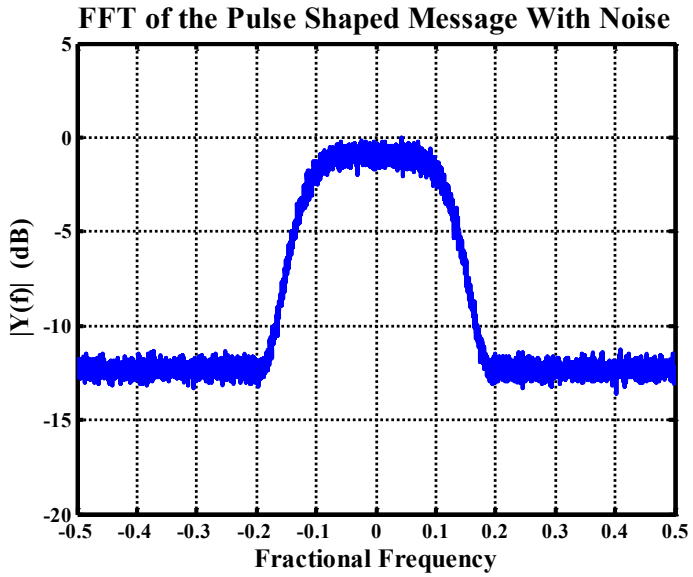

Figure 7: FFT of the filtered message with noise.

3.4.5 Path Loss

To simplify the simulations, the UDNS assumes that the path loss between all transmitters and all SUs is the same. This is a reasonable assumption if all of the SUs are stationary and relatively close together. However, the effective path loss between the each PU and the SUs is simulated, and is taken into account by the SNR of each PU in its channel and the noise floor of each channel.

Later revisions of the UDNS can include path loss in these cases. But having each SU experience the same channel conditions simplifies the problem of classifying the user properties. Removing the path loss from the equation also reduces the number of times the channel simulations need to be run, which increases the speed of the simulations, and allows for more rapid testing.

3.5 Signal Detection

The signal detection used in the UDNS is based on the energy detector discussed in [30], which will be referred to as Cabric's energy detector. This method of signal detection was chosen because it is non-complex and high-speed. The following two subsections will describe the properties of the detector, as well as its use.

3.5.1 Cabric's Energy Detector

Cabric's energy detector averages the N squares of samples to form a test statistic. The energy statistic is defined by

$$\boldsymbol{T = \frac{1}{N}\sum_{N}[x(n)]^2} \tag{14}$$

where T is the energy statistic and x(n) is the series of observed samples. This test statistic is then compared to a calculated threshold to determine the presence or absence of a signal in the band of interest. If an observed test statistic is greater than the threshold, this will cause the detector to declare the presence of a signal.

When a small number of samples are averaged by the detector, Cabric states that the test statistic will exhibit a chi-square distribution with N-degrees of freedom, where N is the number of samples averaged by the detector. Figure 8 shows the distribution of test statistics for a 16-sample detector over 100,000 independent runs, which approximates a chi-square distribution. This agrees with Cabric's findings in her paper, verifying that the energy detector and noise implementation is properly implemented within the UDNS.

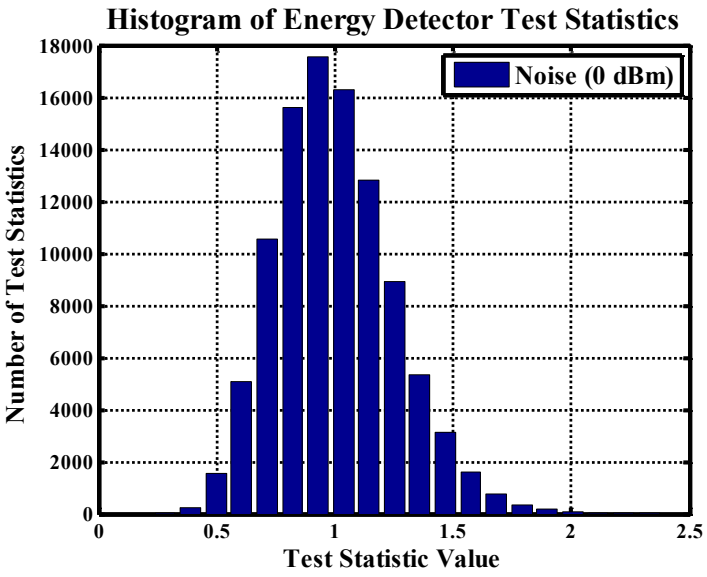

Figure 8: Histogram of the energy detector with 16 input samples.

However, because the chi-square distribution is the sum of N independent variables, the central limit theorem can be used to approximate the test statistic as Gaussian. According to [30], the threshold for the detector to achieve Gaussian distribution is an input length of 250 input samples.

Figure 9 shows the result of a Monte Carlo simulation with 100,000 inpdependant runs, where the distrbution of the energy statistics exibit Gaussian statistics for an input length of 256 samples, a noise power of 0 dBm, and an SNR of -2 dB. The distribution on the left is the test statistic distribution of the noise only, and the distribution on the right is the test statistic distribution of the signal plus noise. The vertical green line at a test statistic of approximately 1.35 is a decision threshold, calculated for a chosen probability of detection and probability of false alarm.

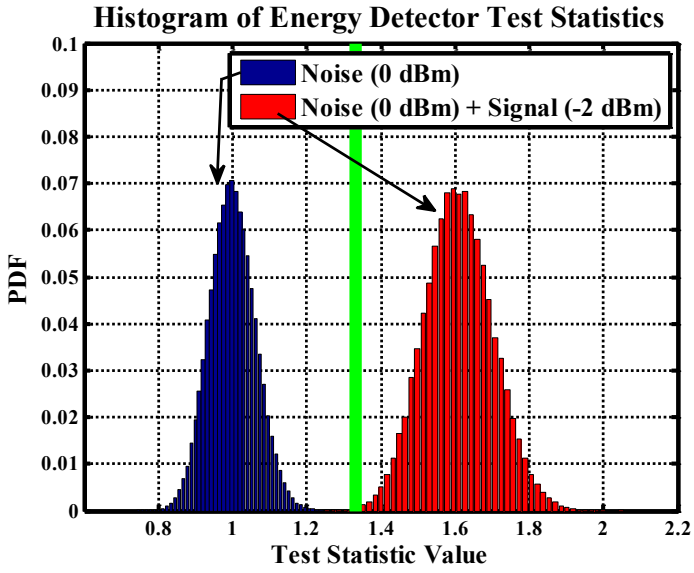

Figure 9: Histogram of the energy detector test statistics with 256 input samples.

Now that the test statistic distribution can be considered Gaussian, the probability of a false alarm and probability of detection can be calculated using q-functions. The functions to calculate the probability of a false alarm and probability of detection are defined in [30] as

$$P_{fa} = Q\left(\frac{\gamma - N\sigma_w^2}{\sqrt{2N(\sigma_w^2)^2}}\right) \tag{15}$$

$$P_d = 1 - Q\left(\frac{\gamma - N(\sigma_w^2 + \sigma_x^2)}{\sqrt{2N(\sigma_w^2 + \sigma_x^2)^2}}\right) \tag{16}$$

where γ is the decision threshold, N is the number of samples, $\sigma_w^2$is the variance of the channel AWGN, and σ_x^2 is the variance of the standard deviation of the signal.

To calculate the detection threshold, either a probability of false alarm or probability of detection is chosen. The probabilities are then calculated over a range of γ, and the γ which corresponds to the correct probability is chosen as the detection threshold. Plotting the two probabilities produces an operating curve for the particular detector, which Figure 10 uses

256 samples. The radio conditions used in calculating the curves are a channel noise power of 0 dBm, for radios with SNR values of -2 dB and 9 dB.

Inside the UDNS, the probability of false alarm is set at 0.1 %. This leads to a γ value of 1.332, which is shown in Figure 9 as the green vertical line, illustrating the relationship of the decision threshold to pure noise (blue) and a low-SNR signal plus noise (red). The use of the false alarm probability rather than the probability of detection is used to make the test statistic threshold independent of the expected signal power.

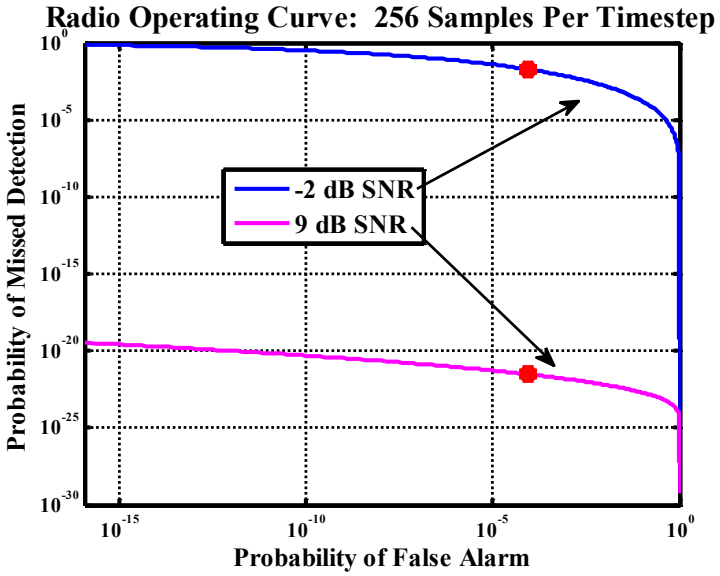

Figure 10: Radio operating curves for 256 samples

Figure 10 compares the operating curves under two different scenarios. The blue line represents a low-SNR secondary user, the magenta line represents a higher-SNR primary user, and the red dots represent where the chosen decision threshold (calculated above) would fall on each curve. Here one of the limitations of this particular energy detector can be seen. This style of detector must be set to a hard decision threshold, and it can either be set to a higher value (moving the red points in Figure 10 to the left along the curves) to reduce the probability of a false alarm, but this would increase the probability that a low-SNR user would

be missed. Or the decision threshold can be set to a lower value (moving the red points in Figure 10 to the right along the curves) to increase the likelihood of detecting low-SNR users, but at the cost of increasing the false alarm rate.

3.5.2 Energy Detector Use

At the beginning of each simulation run, the energy detector is initialized by calculating the decision thresholds. Each SU calculates its own decision threshold for each channel in its channel monitoring list. A different threshold is needed for each channel since each channel has a different noise power.

The energy detector can also be activated and deactivated, either partially or completely. The two functions, which can be activated or deactivated independently, are sensing before transmit and sensing during transmit. The first determines if a channel is currently occupied, and the second monitors the channel on which the SU is transmitting for the resurgence of a PU. Changing these parameters, the effects of a wider range of SU conditions within the system can be evaluated.

3.5.3 Implementation of the Energy Detector

In the UDNS, the energy detector for each SU is active for all timesteps for all of the channels that each SU is monitoring. To simplify the simulator, it is assumed that the SUs in the network possess the capability to simultaneously transmit and monitor on the same channel. This allows the SUs to evacuate a channel immediately before interfering with a PU. However, this method of sensing the same frequencies as one is transmitting on is not physically possible, and is only done as a simplification.

In a real system which uses a time-slotted transmit scheme, the SUs would deactivate their transmissions during the first period of each time block to detect if a PU has begun transmitting. If no PU is present during the first window of the time slot, then the SU would begin to transmit. This would reduce the data rate that the SU could achieve, but would ensure that the SUs do not interfere with the PUs.

In this scheme, there could be issues with two SUs attempting to occupy the same unoccupied frequency and time slot, and not sensing each other's messages within the sensing window. The UDNS resolves this by having the SUs begin transmitting in order of their SU ID number. This gives earlier number SUs priority. In a realistic system, a randomized look-through period would achieve the same effect, allowing SU with the shortest look-through window to transmit, and those with longer windows would be able to sense that SU and choose a different channel to avoid interference between the SUs.

3.5.4 Sub-timestep Signal Detection

While the energy detector implemented in the UDNS operates only on complete timesteps, the detector would be useful in detecting signals prior to a full timestep after the PU has begun transmitting. Equation (16) can be modified to determine the minimum number of samples at which a SU can detect the resurgence of a PU for a given probability of detection.

Since we will be using Equation (16) below a length where the distribution of the test statistics can be approximated as Gaussian, the result will only provide a lower bound on detection performance. The adjusted lower-bound probability of detection equation is

$$\boldsymbol{P_d = 1 - Q\left(\frac{\gamma R - NR(\sigma_w^2+\sigma_x^2)}{\sqrt{2NR(\sigma_w^2+\sigma_x^2)^2}}\right) = 1 - Q\left(\sqrt{R}\frac{\gamma - N(\sigma_w^2+\sigma_x^2)}{\sqrt{2N(\sigma_w^2+\sigma_x^2)^2}}\right)} \quad \textbf{(17)}$$

where γ is the decision threshold, N is the original number of samples, σ_w^2 is the variance of the channel additive white Gaussian noise (AWGN), σ_x^2 is the signal variance, and R is the ratio of the number of timesteps where the SU signal is present to the energy detector input length. The threshold is scaled by R to maintain

Equation (17 can then be plotted against a range of R for an already selected γ value and given noise and signal powers. For this example, the noise power is fixed at 0 dBm and the SNR of the signal varies from 0 dB to 12 dB in 3 dB increments, as shown in Figure 11. However, since the performance difference compared to the full frame is so small, the probability of missed detection is plotted on a semi-log scale.

This illustrates the ability of the SUs to detect the presence of a PU in the channel as a function of the fraction of the frame that the SU uses to test the channel. For realistic channel powers this shows that the SUs can detect the presence of a PU reliable using only a small portion of an already short frame. This supports the sub-frame detection method mentioned in Section 3.5.3.

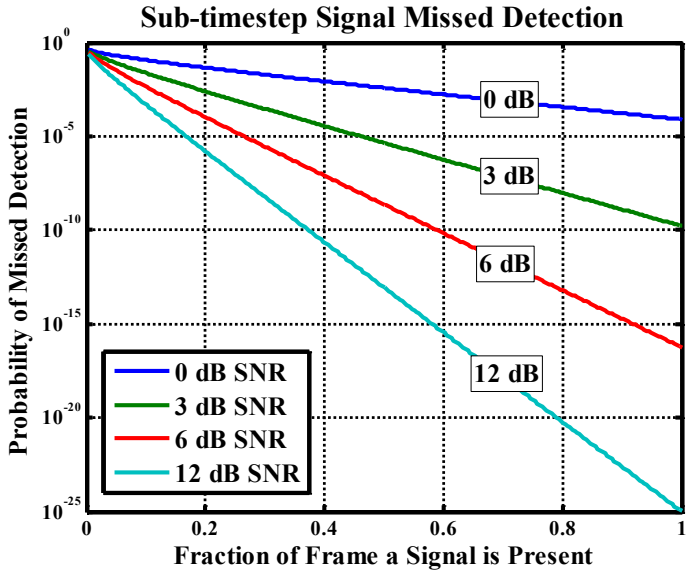

Figure 11: Sub-timestep probability of signal detection.

To confirm the results shown in Figure 11, a simulation was conducted using 100,000 independent runs to test the result for an signal power of 0 dBm, a nose power of 0 dBm, and the γ value of 1.332 from an earlier section. From Figure 11, it is expected that the energy detector would achieve a probability of detection of approximately 80 percent when using 13 of the 256 samples, or 5.08 percent of the frame. Figure 12 shows the chi-squared distribution of the test statistics from the Monte Carlo simulation and the threshold γ, shown as a green vertical line. This gives a probability of detection of 93 percent, which is notably higher than the Gaussian approximation. This further supports the ability of SUs to use a sub-frame sensing method.

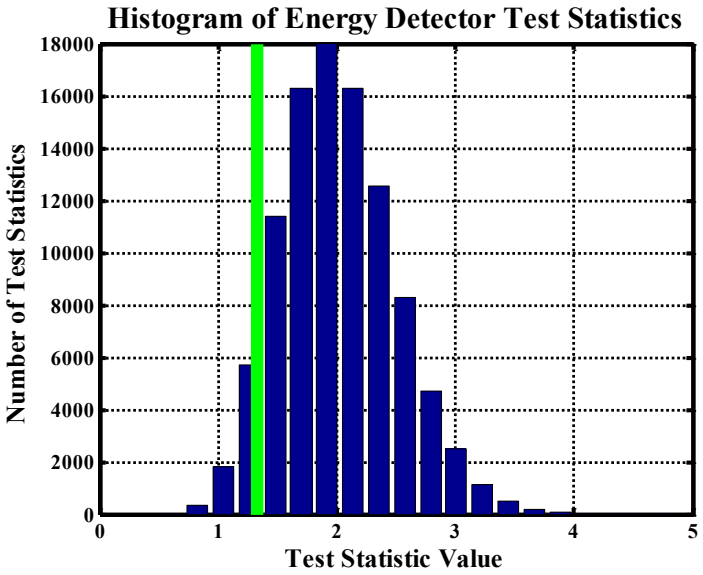

Figure 12: Histogram of ED test statistics for sub-timestep detection – 0 dB SNR.

3.6 Data Recording

During a simulation run, all of the simulation data is saved for use and analysis after the simulation is complete. This data can be used for debugging, simulation analysis, and for the classification methods discussed in later chapters. This is especially useful for the UDNS, because the system currently does not support analysis during an active simulation run. This

also supports continuing simulations at a later date, using the saved statuses. Table 1 shows all of the output variable descriptions. The descriptions have been shortened to reduce the size of the table, but each is discussed in length in the following paragraphs.

Output Variable Descriptions	
Channel occupancies at all timesteps	SU states at each timestep: records the channel decisions and error codes
Radio properties	Channel properties
Current timestep	Current number of channel decisions made by SUs
Channel statistics for each channel decision made by a SU	Radio statistics for the SU that made the channel decision

Table 1: Simulation Outputs

The channel occupancies show the occupancy state for each channel for each timestep. This allows the user to view how the channel occupancy changed as a function of time, such as which channels were occupied, and which radio is transmitting on the occupied channels.

The SU states include any error or warning codes that indicate an issue with each SU. These codes can be found in Table 2. These code flags reflect each covered error state for the SUs. The SU states also contain the channel decisions and channel occupancy of the SUs at all timesteps, and are differentiated from the codes by their sign. Error/warning codes are negative and channel decisions/occupancy is positive.

Code Number	Code Condition
-1	Channel evacuation by PU resurgence
-2	Channel evacuation due to false alarm
-3	Missed detection of SU
-4	SU cannot make channel decision: all channels currently occupied
-5	Unknown error
-6	Channel evacuation by SU transmitting on channel
-9	Missed detection of PU: SU continues transmitting

Table 2: List of SU Status Codes

The radio and channel properties that are output from the simulator contain the final states of their respective features so the simulation can be continued from its final state at a later time. The current step number can be used in a future simulation in continuing the channel occupancy and SU states matrices, or it can be reset to one and those matrices started from scratch. Resetting the step number and SU status matrices has no effect of the outcome of the continuance of the simulation since these matrices are only used for analysis and debugging.

The channel statistics observed by the hypothetical base station for each SU when that radio makes a channel decision are recorded. The channel statistics include the channel energy for that timestep, the channel occupancy rate, the number of timesteps since that channel was last occupied, and the channel access rate. These channel statistics are saved as a four by nChannels matrix, where nChannels is the number of channels, and are organized such that the statistics of the channel that was chosen are moved to the furthest right-hand column. The rest of the channel statistics are organized in order of increasing channel numbers, excluding the selected channel.

For example, if channel three was selected, the channel statistics saved by the UDNS would then be found in the order of [1 2 4 5 6 7 8 9 10 3], assuming ten channels, where the numbers represent the channel number and each position is a column. Likewise, if channel nine was selected, the channel statistics order would be in the order of [1 2 3 4 5 6 7 8 10 9], where the numbers inside the brackets represent the channel statistics for that channel number. They are saved in this way for classification, which is covered in following chapters.

The radio statistics are recorded at each channel decision, and include any statistics that are not included in the channel statistics. Calling these radio statistics is a little bit of a misnomer, but serves the purpose of differentiating this set of statistics from the channel statistics. Table 3 shows the contents of the radio statistics.

Column Number	Statistic Recorded
1	SU ID number
2	Selected channel number
3	SU's channel selection method index number
4	Number of timesteps since the last channel decision by any SU
5	Timestep number of the channel decision

Table 3: Radio Statistics Recorded

The channel and radio statistics are used in conjunction with one another for classification. However, instead of being indexed by the current timestep value, these are indexed by the number of channel decisions made by all SUs in the network for ease of indexing during the classification stage. The channel decision index is the final output of the system, which includes the time at which each channel selection decision in number of timesteps.

4

Experimentation Profile

Chapters 5 and 6 analyze the effectiveness of classification for two sets of objectives. However, before the classification methods are discussed, the standard simulation profile used in the classification experiments needs to be defined, which is the topic of this section.

4.1 Standard Simulation Profile

It is important to test different methods of classification under a set of controlled conditions. This lead to the development of the standard simulation profile. Later chapters cover the effects of modifying the standard simulation profile to support any future work on the topic.

The standard simulation profile controls the initial conditions of the system, including the number of PUs, the PU properties, the number of channels, and the SNR of the SUs. All of the radios in the standard simulation profile share the same sample rate, bit rate, number of bits per symbol, and RRC rolloff factor. When varying the standard simulation profile for classification experiments, the number of SUs and their channel selection strategies were the largest variation used in the classification testing.

The standard profile uses a sample rate and number of samples per timestep, which are shown in Table 4, and are user-definable. Within the UDNS, the code is optimized for timestep sample lengths which are equal to 2^N, where N is any arbitrary integer.

Table 5 shows the standard PU properties that are used in all simulations discussed in this thesis. Most importantly, this table shows that each PU is assigned to a different channel. This table also shows that while some of the properties are fixed, such as the bit rate, rolloff, and bits per symbol. These properties are fixed to reduce the complexity of the simulations.

Sample rate	1 MHz
Number of samples per timestep	256 samples

Table 4: Universal Simulation Values, Sample Rate and Timestep Duration in Samples

	Activity lambda	Idle lambda	SNR (dB)	Assigned Channel	Bit rate (kbps)	Rolloff factor	Bits per symbol
PU 1	30	200	8	1	5	0.5	2
PU 2	20	200	10	2	5	0.5	2
PU 3	60	200	5	3	5	0.5	2
PU 4	50	200	12	4	5	0.5	2
PU 5	100	200	8	5	5	0.5	2
PU 6	90	200	9	6	5	0.5	2
PU 7	85	200	7	7	5	0.5	2
PU 8	60	200	11	8	5	0.5	2
PU 9	80	200	6	9	5	0.5	2
PU 10	20	200	9	10	5	0.5	2

Table 5: Standard Simulation PU Properties

Table 6 shows the standard SU properties. The most important column of the table shows each SUs channel selection method, denoted by its channel selection index number. All of these properties were used for the visualization of a simulation run in the following section. Any value followed by an asterisk is one which is changed during the classification method testing.

	Activity lambda	Idle lambda	SNR* (dB)	Channel selection index*	Channel selection lambda*	Bit rate (kHz)	Rolloff factor	Bits per symbol
SU 1	20	20	10*	1*	inf*	5	0.5	2
SU 2	20	20	10*	2*	inf*	5	0.5	2
SU 3	20	20	10*	3*	inf*	5	0.5	2
SU 4	20	20	10*	4*	inf*	5	0.5	2
SU 5	20	20	10*	5*	inf*	5	0.5	2
SU 6	20	20	10*	6*	inf*	5	0.5	2

Table 6: Standard Simulation SU Properties, with * Marking Values that were Modified During Classification Testing

The SUs have the additional properties of which channels the SU monitors and the channel selection methods that the SU can choose from when the channel selection method counter, which decrements one every time the SU makes a channel decision, reaches zero. The standard operation has the SUs monitoring all of the channels, and the channel selection methods that can be chosen are only index numbers 3, 4, 5, and 6. These correspond to the methods of least energy, least recently occupied, least frequently accessed, and least occupied respectively. Minimum numerically and random selection, index numbers 1 and 2, are rarely used.

The channel selection methods used during the three and four SU tests in chapter 5 are assigned as shown in Table 7. Here, an X is used to indicate that the SU listed on that row is a member of that simulation.

	Three SU Simulation	Four SU Simulation	Channel Selection Method Index
SU 11	X	X	4
SU 12	X	X	5
SU 13	X	X	6
SU 14		X	3

Table 7: SU Channel Selection Methods for Three and Four SU Simulations

The final properties that are a part of the standard simulation profile are the channel noise powers, which can be found in Table 8. These are used to generate the AWGN when the channel is simulated.

	Channel Number									
	1	2	3	4	5	6	7	8	9	10
Noise Power (dBm)	-1	-2	0	-2	1	2	0	0	1	-3

Table 8: Standard Simulation Channel Noise Powers

4.2 Visualization of a Simulation Run

The goal of this section is to show the results of a single, full simulation using the standard simulation profile detailed above. A single simulation was chosen over a Monte Carlo approach to highlight variations between individual simulation runs later in the chapter.

The simulation was run for a total of 5,000 channel decisions by the SUs, which occurred over 35,026 timesteps. This simulation took 553 seconds to run and represents approximately 8.96 seconds of simulated time. Simulations with these parameters take an average of 555 seconds to complete, with a standard deviation of 9.3 seconds.

4.2.1 Primary Users

To verify the implementation of the activity and idle lambda, the occupancy rates for each channel are reviewed, and are shown in Figure 13. From Table 5 PU 2 has activity and idle lambdas of 20 and 200 respectively, which correspond to an expected channel occupancy rate of nine percent. Likewise, PU 5 has activity and idle lambdas of 100 and 200, which corresponds to an expected channel occupancy rate of 33 percent. Figure 13 shows that PU 2 had an actual channel occupancy rate of 9.1 percent, and PU 10 had an actual channel occupancy rate of 32.6 percent, which are within experimental error to the expected results.

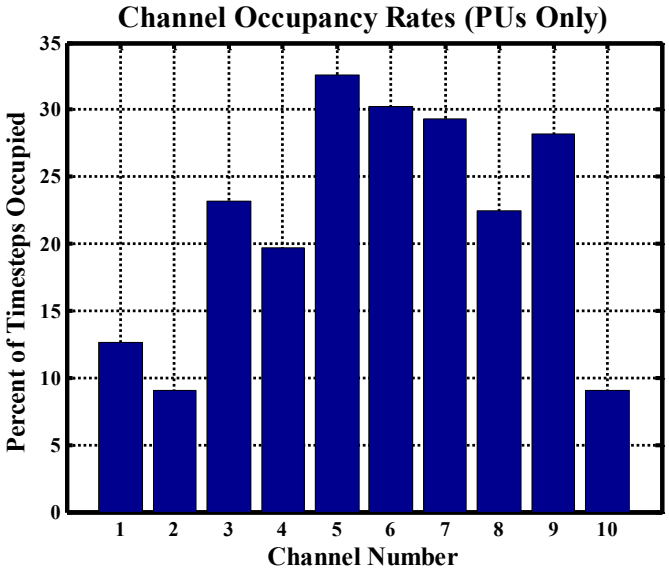

Figure 13: Channel occupancy rate for PUs only.

4.2.2 Secondary Users

For this simulation, six SUs were used. The SUs use the channel selection index equal to their SU ID number. Therefore, SU1 uses channel selection method one, and SU 4 uses channel selection method four.

Comparing the results in Figure 13 to the results in Figure 14 shows a significant increase in the use of the channels, and is one of the reasons DSA networks are being researched. That is because they can be effective at increasing the use of spectrum by decreasing the amount of downtime in existing channels. The contribution of SUs to the channel occupancies can be seen in green in Figure 13.

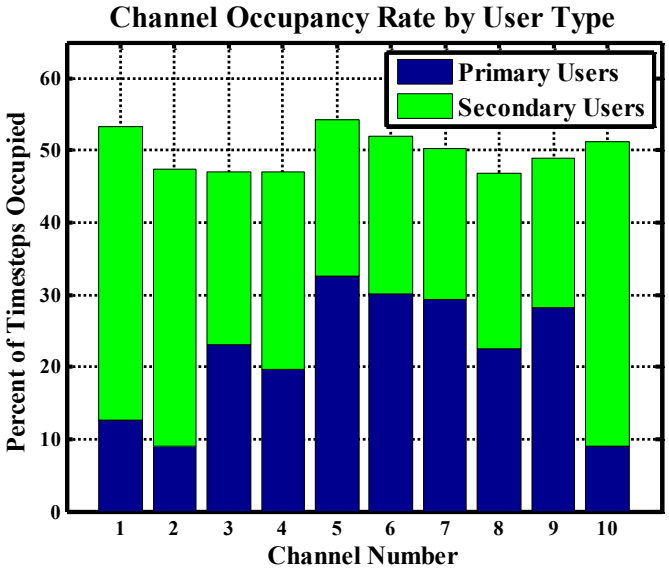

Figure 14: Channel occupancy rates for all radios.

In this example, the addition of the six SUs increases the use of the spectrum by almost 35 percent over the PU-only case. In the model used by the UDNS, the SUs are unaware of the actions of the PUs until the energy detector built-in to the SUs detects that a PU is transmitting. In this model, there is an inherent amount of interference. However, this interference is limited to the first timestep, during which the SU detects the presence of the

PU's signal using the energy detector described above. Figure 15 shows the percent of the number of timesteps that the SUs interfered with the PUs, broken down by each channel/PU.

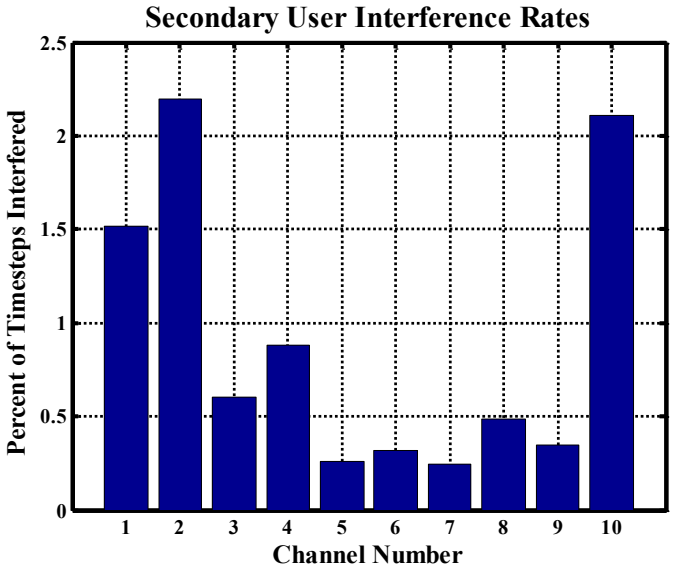

Figure 15: SU interference rates by channel number.

For a realistic system, any amount of interference by the SUs is unacceptable. This is why, in a realistic system, the look-through method introduced in Section 3.5.3. As mentioned, there would be a small, but tolerable, reduction in throughput from the SUs.

Additionally, the interference in this model is dependent on the average number of timesteps that the PU transmits during each transmit cycle. PUs with large activity lambdas, such as PUs 5 and 6 which remain active for longer periods of time, decreases the impact of SU interference during the first timestep on the entire message.

4.2.3 Channel Accesses per Secondary User

A feature that is assumed in the classification testing is that all radios make approximately the same number of channel accesses; and therefore, experience roughly the same number of channel selections. This is desired because equalizing the number of channel

selections made by each user allows the classification results to the relative merits of each method, without weighting due to better estimation of one type of channel selection method.

Initially, it was expected that the number of channel accesses by each user in a longer simulation should be approximately the same. However, adding in the effects of channel evacuations and subsequent channel re-selection causes some SUs to have more channel accesses than others. Figure 16 shows the number of channel accesses broken down by the SU number for the 5,000 channel decision simulation run. This shows that a simulation run of 5,000 or more is sufficient to achieve an approximately constant number of channel decisions per secondary user.

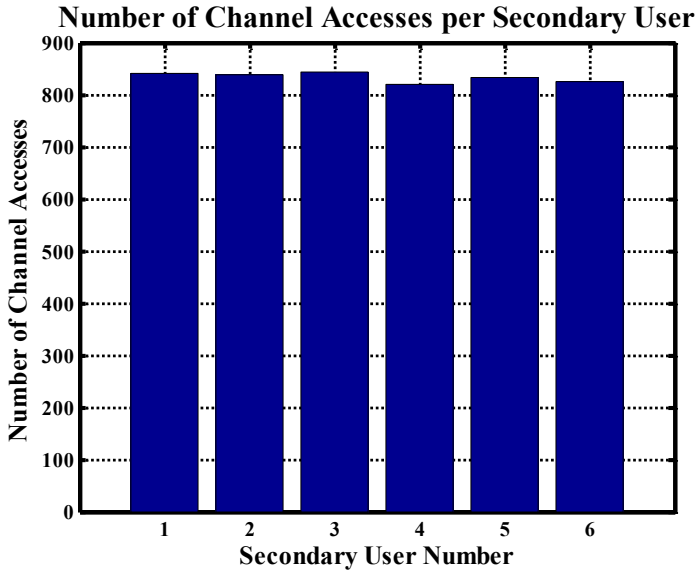

Figure 16: Channel access count by user.

4.2.4 SU Channel Selections

The next step is to look at the results of the channel selections made by the users according to their channel selection strategies. In an ideal situation, the channel selection methods would operate without constraints imposed by the presence of other users already on

the channels. However, the SUs must not interfere with the PUs on their channels or with other SUs which have already begun transmitting on a channel.

Figure 17 shows the number of channel selections made by each radio during a simulation run using the standard simulation profile. As the figure shows, most channel selection strategies have a favored channel on which it transmits most often. This is because the stationary channel statistics meets the requirements of that SUs channel selection method. However, that channel is not always available to the SU. An example of this is the minimum numerically method. It would prefer channel 1, but due to previously present signals, it is pushed to select higher-number channels 15 percent of the time.

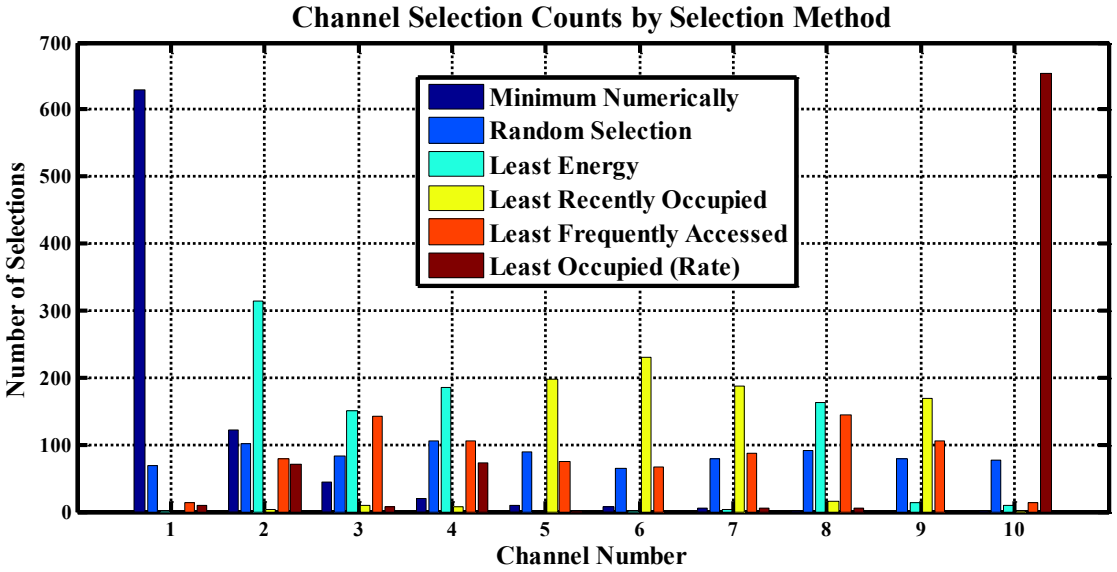

Figure 17: Count of channel selections by channel selection method.

Additionally, the channel selections are affected by the reemergence of the PUs on their channels while the SUs are transmitting. Some of the channel selection methods are more effective at selecting channels than others, which can be seen in their amount of time transmitting compared to the number of channel evacuations that the SU was forced to make.

These results are in Figure 18. In this case, a higher number is better, as it corresponds to a greater amount of time spent transmitting on each channel before a channel evacuation is needed. This is a fairly surprising result, since minimum numerically (#1) and random selection (#2) perform better than least energy (#3).

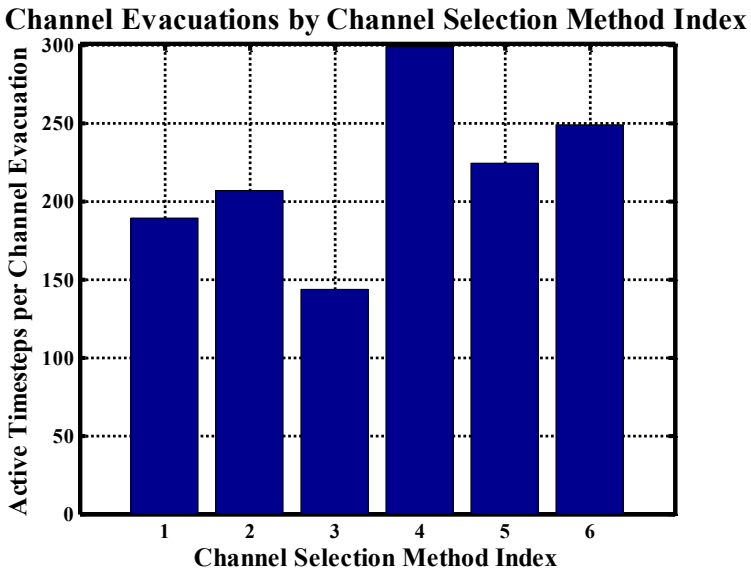

Figure 18: Channel evacuation properties by channel selection method.

5

Classification of Secondary User ID

The standard simulation profile will now be used to perform tests on classification accuracy. The first method of classification investigated is the use of the Naïve Bayesian classifier to quickly estimate the user ID number of a SU that begins transmitting.

5.1 Motivation

The first classification method is motivated by the need by a SU to be able to estimate the other SUs operating in its broadcast area. However, it would be computationally expensive for the SU to receive and decode all channels simultaneously to determine what user is transmitting. Therefore, this chapter will cover the estimation of SU ID number by a hypothetical observer, with the same sensing equipment as a SU.

5.2 Classification Objectives

The hypothetical observer will estimate the radio ID of a set of N SUs when given only the channel conditions observed at the time of their channel decision. This is useful, as it allows an observer to quickly estimate the ID of radios without more complex equipment, as the method will only rely on channel monitoring equipment that would be included in a standard SU within the DSA network. However, to train the classifier, it is assumed that the

observer has the ability to differentiate between the SUs, using a method such as detecting the look-through periods used by the SUs, which are not present in the PUs transmissions.

5.2.1 The Classifier: NaiveBayes

To increase the speed of the processing with a large number of input data values, the Naïve Bayesian classifier described in Section 2.3 is used for the classification. These tests use the Matlab function NaiveBayes, which is built into the Matlab Statistics toolbox. This function has several different distribution options for the inputs.

For this thesis, the distribution chosen was the normal distribution ('normal') out of the options of kernel ('kernel'), multinomial ('mv'), and multivariate multinomial ('mvmn'), where the Matlab tag for each is noted within the parenthesis. The properties for these distributions can be found in Table 9.

	Description
'normal'	• Features within each class have normal distributions, and are described by their mean and standard deviation • Assumes continuous distribution.
'kernel'	• Does not assume normal distributions • Computes a kernel density estimate for each class using the training data • Useful if data skewed with multiple peaks • Assumes continuous distribution.
'mv'	• Features represent counts of words or tokens • Useful for spam filtering
'mvmn'	• Useful for categorical features

Table 9: Distributions Built in to the Matlab NaiveBayes Function

Some tests involving the kernel distribution were conducted. However, the accuracy gains presented were minimal and came with a significant computational cost. In this thesis,

which implements the Naïve Bayesian classifier partially due to its speed, this increase was not tolerable. In tests conducted using the negative flagging method described Section 5.5, using the kernel distribution compared to the normal distribution for the same test parameters increased processing time approximately 325 times.

5.3 Classification Method: Direct Classification

The method of classification first investigated involves passing the entire channel data set, directly from the saved values recorded by the UDNS during the simulation, into the classifier for training and estimation. This method will be referred to as direct classification because there is no processing of the channel statistics before passing into the classifier. The data passed includes the channel energy, the channel occupancy rate, the number of timesteps since the channel was last occupied, and the channel access rate for each channel in the network. In all of the tests conducted in this chapter, the SUs monitor all of the channels in the network.

To train the classifier, we determine a reference point within the sample space defined by the series of recorded channel decisions by the SUs. This reference point is used to determine which sets of channel data are used for training and classification, with channel selections occurring before the reference point being used for training the classifier, and channel selections occurring after the reference point being used for estimation. For the following plots, the classifier was trained using all channel selections occurring prior to the reference point, which is referenced on the x-axis, and then estimates the following 200 channel decisions using the channel properties experienced at each channel decision.

Figure 19 shows a visualization of the classification process, with the classifier and training phases being simplified to input/output blocks. The green sets of data represent the training data, the light blue represents the reference location, which is included in the training data, the dark blue represents the channel data used for estimation, and the red represents the estimated user ID values. All of the tests run in this thesis were run after a complete simulation was run, emulating a real-time system. In a true real-time system, the uncolored table entries would represent future time.

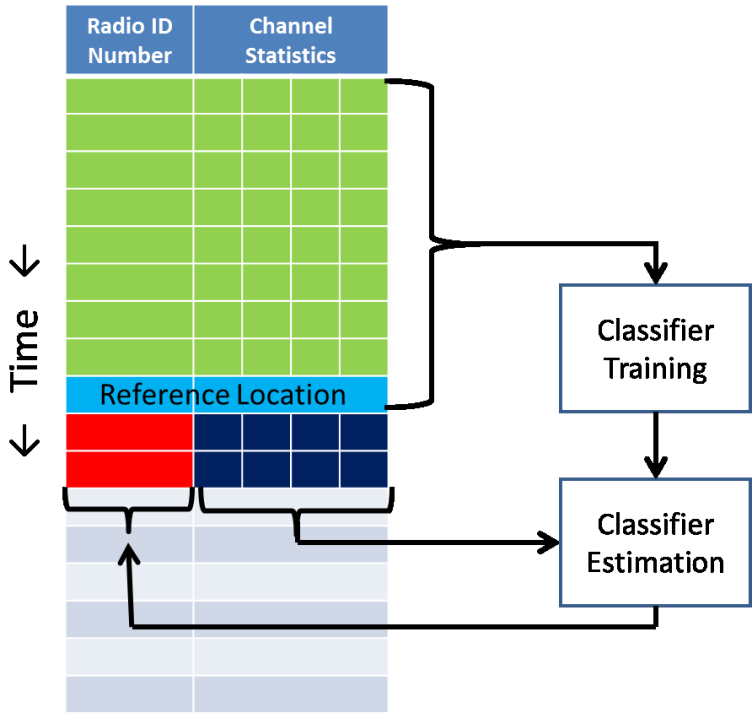

Figure 19: Visualization of the direct classification method.

5.3.1 Train/Test Length

Before the method is tested, the effects of the training and testing lengths on the classification accuracy must be discussed. The first test involves the relationship between the number of training samples and the number of forward and backward predictions. Forward prediction involves estimating the SU ID number for channel selection instances which have

not been used as a part of the training set, whereas backward predictions involves the using the same channel selection instances for both training and estimation.

Figure 20 shows the relationship between the ratio of training to testing lengths and the accuracy of classification. This was tested over 10 independent runs for a fixed training length of 200, and testing lengths of +/- 1, 5, 10, 25, 50, 75, 100, 150, 300, and 450, and three SUs. This figure shows a drop in classification accuracy for forward prediction, but the drop in accuracy is small compared to using the same data points for training and estimation. It also shows that backward prediction accuracy degrades quickly when the testing length is more than half of the training length, but this is not true for forward prediction. The implications of this will be discussed later in this thesis.

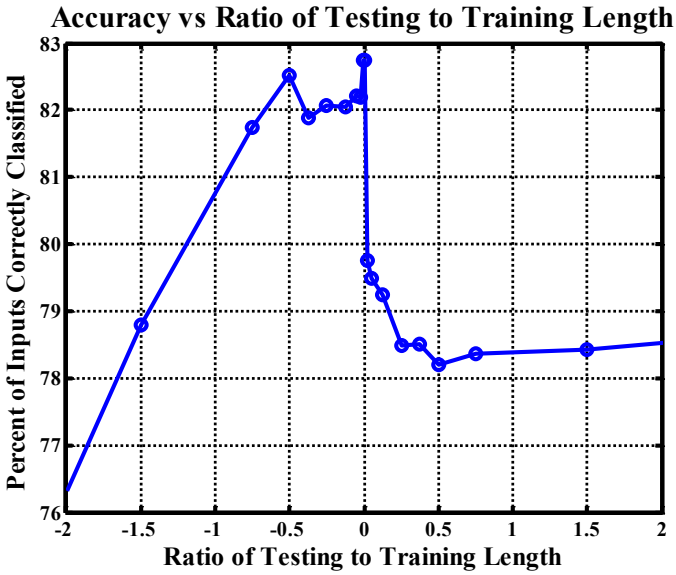

Figure 20: Estimation accuracy against ratio of training to test lengths.

From this point forward, unless otherwise noted, all classification tests will use training and testing lengths which are equal to one another. The standard value which will be used is 200.

5.3.2 Direct Classification Results

Direct classification for three and four SUs in a 10 PU network yields the accuracy plot found in Figure 21. The plot shows a single realization of the accuracy plotted against time. A single realization was chosen to illustrate the volatile nature of the classification accuracy. Using this method, the three SU case yields an average classification accuracy of 76 percent, while the four SU case yields an average classification accuracy of 72 percent. Testing was conducted using testing and training lengths equal to 200.

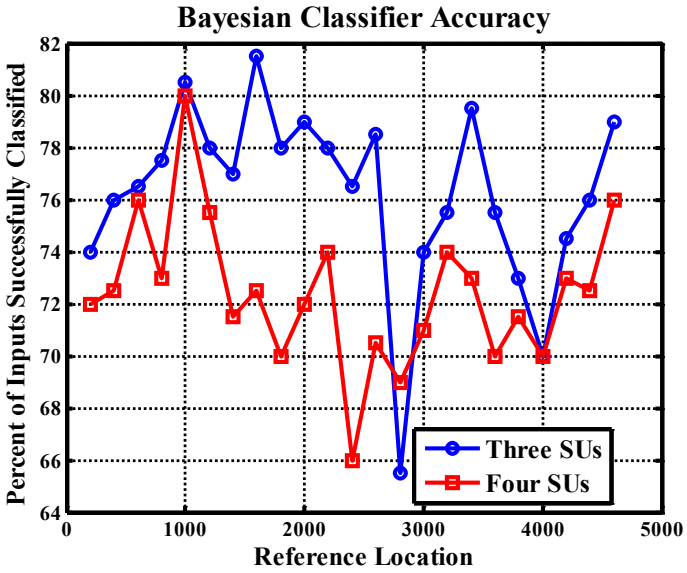

Figure 21: Bayesian classifier accuracy using the direct classification for three and four SUs in a 10 PU network.

5.3.3 Energy Detector Deactivation

One method of determining the source of some of the inaccuracies in the classification is to selectively deactivate the energy detector (ED) for the SUs in the network. This causes the UDNS to produce results which do not accurately simulate a real-world DSA network. However, since this was purely a method of determining sources of error in the classification process, simulation accuracy is not essential.

Particularly, this was a method of testing if the classifier is not being passed, or incorrectly being passed, information about which channels are occupied and open, and the effect of channel evacuations on accuracy. The effect of these can be determined by comparing the accuracy results of the previous method of direct classification to direct classification of the data with the energy detector deactivated.

To determine the effects of channels being occupied by other radios, the energy detector was deactivated before the SUs transmitted. This eliminates the classification problem of some channels being avoided due to occupancy. The second test, to determine the effect of channel evacuations, was calculated by deactivating the SUs energy detector while the SUs transmit. This eliminates the ability of the SUs to detect the resurgence of the PU on their channel, removing PU resurgence as a source of confusion to the classifier.

5.3.4 ED Deactivation Results

Deactivating the energy detector results in a classification accuracy of approximately 83 percent in a four-SU, 10 PU simulation, which is a raw increase of 10 percent in classification accuracy over the full simulation results presented in Figure 21: Bayesian classifier accuracy using the direct classification for three and four SUs in a 10 PU network.. This is a significant increase in accuracy. Meanwhile, deactivating the energy detector during SU transmit for a four-SU simulation has a negligible effect on performance, with the average classification accuracy remaining relatively unchanged.

This is significant since it tells us what information the classifier is not receiving properly. The classification accuracy being affected by the energy detector deactivation shows that the classifier is not being passed proper information about which channels are

currently occupied. Additionally, the lack of effect on classification accuracy with the energy detector deactivation during SU transmission means this is likely not a limiting factor on classification accuracy. The next two sections investigate two methods of passing this information.

5.4 Classification Method: Hard-decision Pre-processing

The first successful method investigated to pass to the classifier which channels are occupied at the time of the channel decision was the application of pre-processing to apply hard-decision rules to the channel statistics prior to classification. This method forces relationships between the inputs and decreases the number of dimensions.

5.4.1 Hard Decision Testing

The pre-processing algorithm involved hard decisions with binary outputs, and is referred to the hard-decision pre-processing method. The hard-decisions were implemented using a series of if-statements. A binary test is run on each channel for each of the four channel selection methods that use the channel statistics to determine which channel to occupy. These channel selection techniques are the least energy, least frequently occupied, least recently occupied, and least accessed. Before testing, any channels that are currently occupied are set to a value of NaN to simplify the testing process.

For each channel selection instance, the statistics of the unoccupied channels are tested if for the four channel selection techniques from the previous paragraph. The binary outputs for the four tests for the selected channel is then stored in a matrix until all of the channel selection methods have been tested for all decision times of interest. This storage matrix has

four columns, one column for each channel selection method, and the same number of rows as the number of channel decisions that have been tested.

At this step, the number of dimensions has been decreased by a factor of 10. A further reduction in the number of dimensions is desired, but classification accuracy will be tested at this point as well. This should result in extremely high accuracy results, because at this point the channel statistics handed to the testing algorithms were the same as those used by the SU, with only a small variation in the channels that are registered as occupied due to false alarm detections by the SU.

The classifier used to do this intermediate classification test was the same Matlab function that is the used for most of the classifications in this thesis. However, that function assumes that the inputs are continuous, and require the within class variance, or variance of each set of X_i values that correspond to $Y=y_k$, be positive. In this test, the X_i corresponding to that SUs channel selection method should always be equal to one. This would violate the within class variance requirement for the classifier, and return an error.

Therefore, prior to classification, a matrix with Gaussian statistics and a standard deviation of 0.00001 is created with a size equal to that of the binary test matrix. The two matrices are added to one another, which ensures that the Matlab function does not throw an error for not having positive in-class variance. Figure 22 shows the plot of the accuracy of classification against the reference location, with fixed testing and training lengths equal to 200.

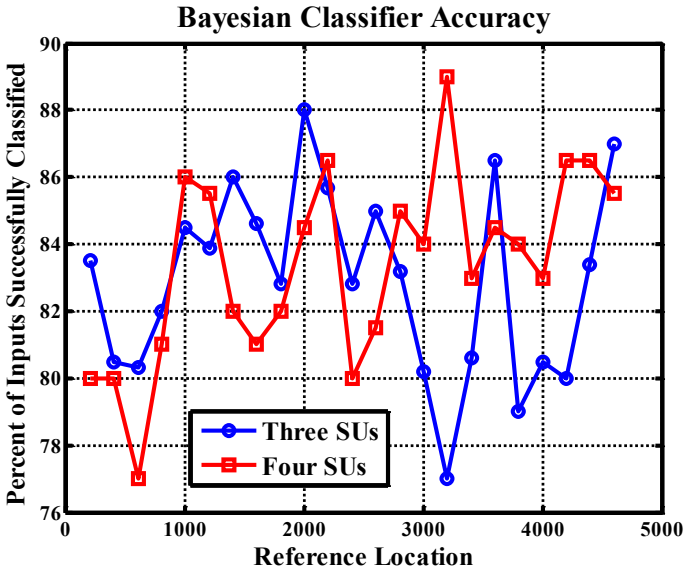

Figure 22: Bayesian classifier accuracy using hard-decisions for three and four SUs in a 10 PU network.

This result shows classification accuracy vey near to the results found in the energy detector tests. Specifically, this method yielded an average accuracy of 83 percent for a four SU case, which is within experimental error of the 84 percent result found in the energy detector deactivation test. This is significant since it shows that the channel data is being passed properly to the classifier. Equally important is the fact that this closes gap in classification accuracy between the three- and four-SU simulations from four percent on average to nearly one percent.

5.4.2 Decision Mapping

After testing and creation of the binary matrix is complete, the four binary inputs are mapped to a two-dimensional space. The chosen method involved the use of nine states. This map is generated using the equation

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{B}(\mathbf{4}) - \boldsymbol{B}(\mathbf{3}) \\ \boldsymbol{y} &= \boldsymbol{B}(\mathbf{2}) - \boldsymbol{B}(\mathbf{1}) \end{aligned} \tag{18}$$

where x and y represent their respective axis, B is the matrix of binary test values, and B(n) is the nth column within B. A diagram of this mapping is shown in Figure 23. However, since a binary number with four bits has 16 potential values, there are ambiguous cases involved in this mapping.

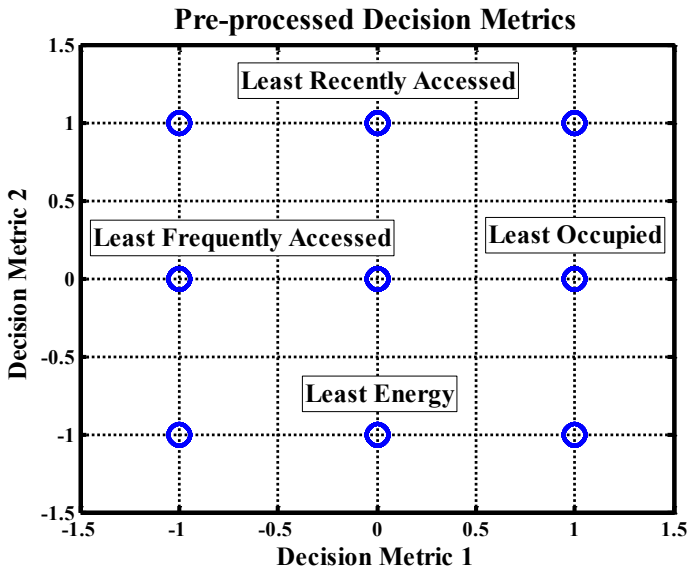

Figure 23: Pre-processed decision metric mapping.

The ambiguous cases occur at the origin, which represents one of four cases, and at the four points directly on the x and y axis, which represent one of two cases. The origin can represent either [0 0 0 0], [1 1 1 1], [1 1 0 0], or [0 0 1 1]. While the points on the axis can represent either [1 0 0 0] or [1 0 1 1], where the position of [1 1], and the position and order of [1 0] can be changed to the proper form for the particular on-axis point. The combination [1 0 1 1] would be located at coordinate (0, -1) on the map.

This two-dimensional set of training and testing data is then passed to the classifier. The results of the classification can be found in Figure 24. This shows an increase in the accuracy of classification compared to the direct classification approach. Specifically, this

yields an average accuracy increase of approximately eight percent for both three and four SU cases when compared to the direct classification approach shown in Figure 21.

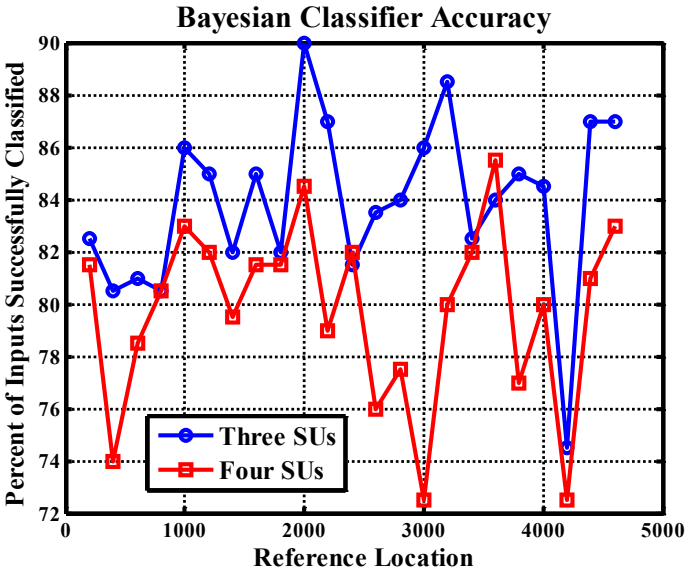

Figure 24: Classifier accuracy using nine hard-decision, pre-processed points.

However, there is the problem of this method not performing well when multiple radios use the same channel selection method due to the limited amount of information being passed to the classifier. And this method requires, in the presence of different or more complex channel selection methods and algorithms, that the observer has knowledge of the operation of each of these possible selection methods. Because of these considerations, a better method of passing this information to the classifier is needed, and is discussed in the following section.

5.5 Classification Method: Negative Flagging

To improve the performance of the classifier, a final approach of 'flagging,' or marking the channel statistics in a way that is meaningful to the classifier, shows which channels are occupied at the time of the channel decision. This method is referred to as the

negative flagging method. After testing a number of methods, the simple application of negative flagging was found to be the most useful and accurate.

Negative flagging involves multiplying the channel statistics for channels that are occupied at the time of a channel decision by -1. This creates a second state within the statistics for each channel that indicates that that channel is currently occupied. This also allows for the additional information of which channel is to be passed to the classifier.

The most significant feature of this method is that it does not make any assumption that the observer knows what channel selection methods are being used by the SUs. This is found to be useful in Chapter 6.

Implementing the negative flagging method is identical to the direct classification method after the statistics for the occupied channels are negated, with the sign-encoded channel statistics being passed directly into the classifier. This results in a large-dimension classification problem, but since the Naïve Bayesian classifier complexity grows linearly, this is not a significant problem when compared to the hard-decision pre-processing method.

5.5.1 Negative Flagging Results

Figure 25 shows the accuracy results for a three- and four-SU simulation using the negative flagging method in a 10 PU network. These simulations use equal training and testing lengths of 200 channel decisions. This shows both radios having approximately the same classification accuracy, 77 and 78 percent for the three and four SU cases, respectively.

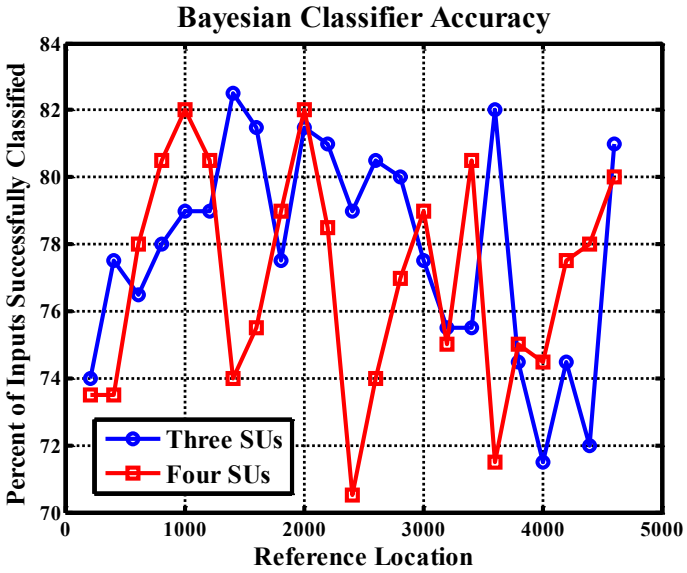

Figure 25: Classifier accuracy using the negative flagging method for three and four SUs in a 10 PU network.

This is a seemingly poor result when the gains are compared to the direct classification method for the three radio case. Especially when compared to the hard-decision methods. However, the important result is the increase in the number of SUs that the negative flagging method can support. Using direct classification, the three- and four-SU cases were separated by an average accuracy gap of four percent. With the negative flagging, the gap has closed to less than one percent, as well as the three-SU case accuracy increasing by two percent overall.

5.5.2 Increasing the Number of SUs

The next test of the performance includes the addition of multiple SUs using the same channel selection methods. Inclusion of the additional radios increases the channel utilization, and therefore decreases the accuracy of classification following the results in Figure 27, which is presented and discussed in Section 635.6. To account for the increase in the number of SUs, the activity lambda for each SU is decreased and the idle lambda increased such that the

overall average active rate for the combination of all SUs remains constant compared to the standard four-SU case.

The activity and idle lambdas are modified from the standard simulation profile according to the equation

$$\boldsymbol{\lambda_{activity}} = \frac{\mathbf{80}}{\boldsymbol{N_{SU}}} \qquad \boldsymbol{\lambda_{idle}} = \mathbf{40} - \boldsymbol{\lambda_{activity}} \tag{19}$$

where $\lambda_{activity}$ is the activity lambda, λ_{idle} is the idle lambda, $\hat{\lambda}_{activity}$ is the new activity lambda, $\hat{\lambda}_{activity}$ is the new idle lambda, and N_{SU} is the number of SUs in the network. Using the new activity and idle lambdas, a simulation was run using eight SUs. All four of the channel selection methods are implemented, with two radios using each channel selection method. The resulting radio properties can be found in Table 10.

	Channel Selection Index	Activity Lambda	Idle Lambda
SU 1	4	10	30
SU 2	4	10	30
SU 3	5	10	30
SU 4	5	10	30
SU 5	6	10	30
SU 6	6	10	30
SU 7	3	10	30
SU 8	3	10	30

Table 10: Eight SU Simulation Properties

Figure 26 shows the accuracy of the results using the negative flagging method for the four- and eight-SU simulations. This shows approximately a 20 percent decrease in the classification accuracy when doubling the number of SUs, without increasing the size of the PU network. One reason for this reduction in classification accuracy is confusion between radios using the same classification method. This is tested and covered more extensively in Chapter 6.

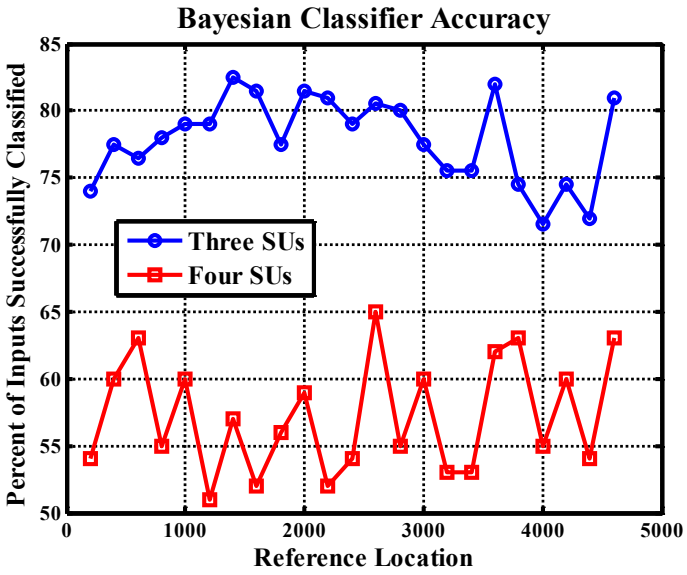

Figure 26: Classification accuracy of four and eight SUs in a 10 PU network.

5.6 Accuracy and Channel Utilization

Testing the classification accuracy across channel utilizations, it was found that in a moderately-utilized network, the accuracy of classification begins to drop off dramatically. For this thesis, a moderately-utilized network is defined as one which has an average utilization of over 45 percent after adding the SUs to the network. Figure 27 shows a plot of the accuracy of classification plotted against the average spectrum efficiency for a series of four-SU simulations. After the point where the network is considered to be moderately-utilized, the classification accuracy decreases linearly with the spectrum efficiency.

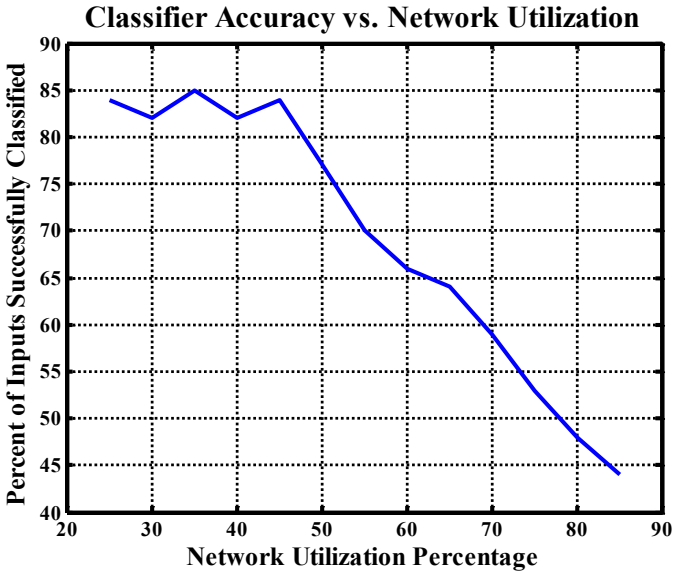

Figure 27: Classification accuracy as a function of spectrum utilization.

For the test run in Figure 27, the spectrum efficiency was controlled by using the idle lambda value for the PUs, which was set at increasing values from 10 to 1500. The activity lambda for the PUs was left equal to their value in the standard simulation profile. After each simulation run, the spectrum efficiency is calculated by dividing the number of timesteps that the network was occupied by the product of the number of timesteps in the simulation and the number of channels. The statistics for the SUs remain unchanged from the standard simulation profile, with four SUs present in the simulations.

The reason the classification accuracy decreases with the spectrum efficiency is due to the probability that a channel decision is made not due to the channel selection method. Rather, the channel selection is made only by selecting the only channel that is unoccupied. In this case, given that all radios have roughly the same probability of occurrence, the classifier can only achieve accuracy equal to that of random selection. And as the spectrum efficiency increases, the probability of this occurring increases linearly.

6

Classification of Channel Selection Method

Moving forward with realistic implementations of DSA networks, a more useful classification objective would be the channel selection methods that each radio, or group of radios, is using. This chapter covers a method of estimating a radio's channel selection method by grouping radios with similar channel selection methods.

6.1 Classification Objectives

Mentioned above, this chapter investigates the usefulness of the Naïve Bayesian classifier in grouping SUs by their channel selection method. This is a very useful ability for SUs in a realistic DSA network. This allows an observer to be able to accurately predict which channel a SU will select when it is time for the user to make a channel decision. If the observer is another SU, it allows that user to better select channels that reduce the probability of interference to other SUs. This increases the channel utilization and reduces the probability of a race condition between two SUs on the same channel.

6.2 Classification Method

Classification of the channel selection methods presents an unusual challenge to the Naïve Bayesian classifier. Specifically, it is the assumption that the observer, and its associated classifier, does not at any point know the channel selection technique that a

particular radio is using. Therefore, a method of training a classifier using the channel selection method's index, or some other imaginary index, cannot be done.

To solve this within the Naïve Bayesian classifier, a method of two-step clustering was instead implemented. The first step is classification of the data using backward analysis, and the second is the determination of similarities.

6.2.1 Multiple SUs using the Same Channel Selection Method

As shown earlier in Figure 20, there is a considerable increase in the classification accuracy when the same data points are used for training and classification. Unlike before, this method is now a viable option. Since we are assuming that the observer knows the ID of the SUs which are transmitting, there is no need to determine this value using the forward prediction of the previous chapter.

Unlike the previous chapter, these tests require higher numbers SUs to realize the usefulness of the grouping. Particularly, we need at least two radios to use the same channel selection method. For the initial test, the 8 SU simulation profile described in Table 10 of the previous chapter is used. This simulation includes eight SUs, with all four of the channel selection methods implemented, where two radios use each channel selection method. Using the negative flagging method, Figure 26 shows that the classification accuracy drastically decreases for larger numbers SUs using the same channel selection method when attempting to estimate the user ID.

To test the hypothesis that this reduction in classification accuracy is due to confusion between radios using the same classification method, a test was designed using the negative flagging method discussed in Section 5.5. However, instead of measuring accuracy by

comparing the user ID numbers, the accuracy is measured by comparing the channel selection method indices. Otherwise the training method is unchanged.

The results of this can be found in Figure 28, and demonstrate that there is a small difference average in classification accuracy between the four-SU and eight-SU test scenario. The four-SU scenario has each SU using a different channel selection method, and the eight-SU has two SUs using each radio selection method. The small difference between the two is likely due to experimental error. This confirms that the drop in classification accuracy when adding multiple SUs with the same channel selection method, seen in Figure 26, was due to the classifier confusing the radios.

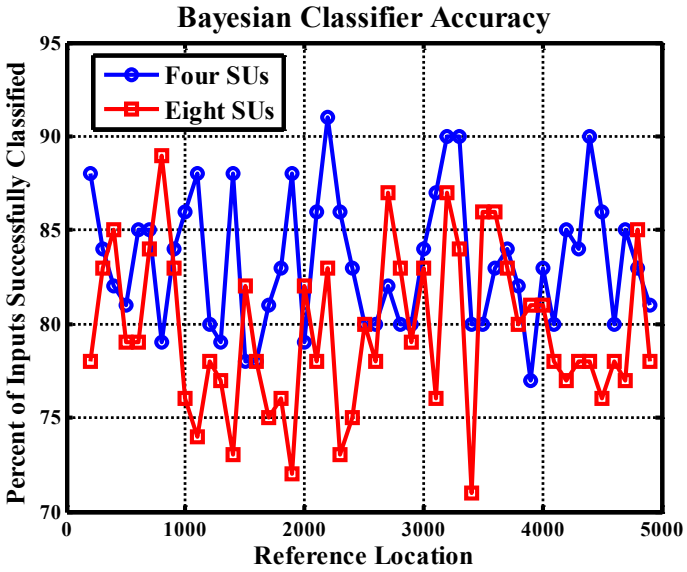

Figure 28: Classification accuracy for four and eight SUs when determining accuracy of estimating the proper channel selection method.

This is an important result because it shows that the classifier is often correctly classifying the channel selection methods. How we make this useful is discussed in the following section.

6.2.2 Backward Classification

When running the classifier, the SU ID index is the initial objective of the classification. It is used an intermediate step to determining the clusters. However, since we are no longer trying to forward-predict the SU ID number, we can take advantage of the increased classification accuracy seen in Figure 20 when using the same data for training and classification. We can use this to compare estimated values to the known, correct value for that instance of classifications. This allows for the detection of errors from the classification. These errors are then used to determine the similarities between the SUs, which is the goal of Section 6.3.

When a classification is run, two sets of outputs are used. The first set includes the estimated, most probable value calculated within the classifier. This will be referred to as the classifier output. The second set of values includes the probability of occurrence for each of the possible output states that is calculated within the classifier. These will be referred to as the classifier probabilities. The most probable value is the state with the highest probability of occurrence.

6.3 SU Grouping: Classifier Histogram

The first method of grouping the SUs investigated will be referred to as the histogram approach. This refers to counting the number of occurrences for each classifier output for a given, known state. The idea behind the histogram method is that SUs with similar channel selection statistics will make similar channel decisions. And since the classifier is passed no additional information besides the channel statistics, radios with similar channel decisions

should have a high confusion rate, or number of occurrences where the classifier estimates the wrong SU ID number.

6.3.1 Histogram Setup

Before describing the testing process, Figure 29 shows a histogram of the SU ID sections. This histogram represents the number of times that a given, known result of SU A, shown along the x-axis, was estimated to be the result shown in the vertical colored bars. For example, if SU 18 was the estimated state at channel selection instance *n*, and the known result was SU 15, then the value corresponding to the SU 18 bar in column group 15 is incremented by one.

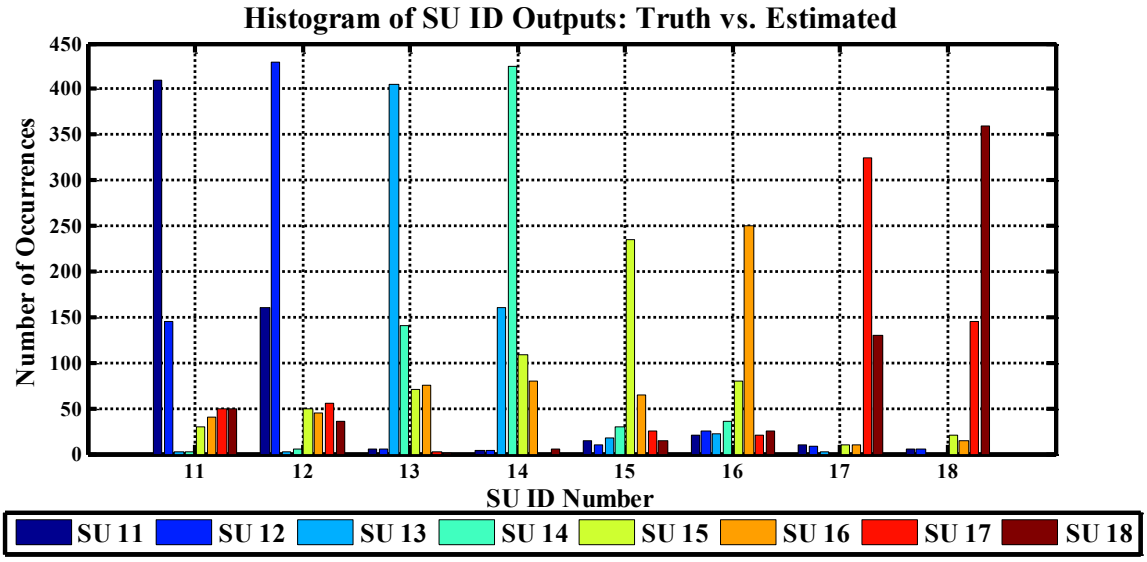

Figure 29: Histogram of SU ID estimations, plotting the number of estimated SU IDs for each truth SU ID over 5,000 channel decisions.

Another useful histogram is the inverse of the previous method. This second histogram displays the estimated SU ID number is shown along the x-axis, and what the actual SU ID number truth is shown in the vertical bars. For example, if SU 18 was the estimated state at channel selection instance *n*, and the known result was SU 15, then the

value corresponding to the SU 15 bar in column group 18 would be incremented by one. This is shown in Figure 30, which is calculated from the same simulation results used to generate Figure 29.

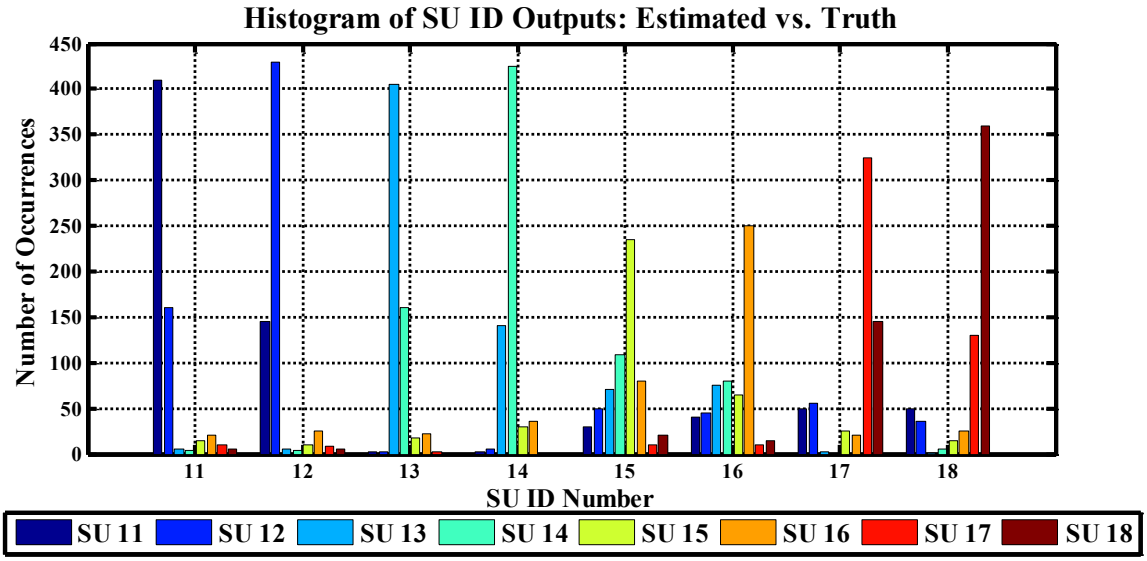

Figure 30: Histogram of SU ID estimations, plotting the number of truth SU IDs for each estimated SU ID over 5,000 channel decisions.

6.3.2 Creating the Similarity Matrix

From these results, a simple method of comparing the similarities between the radios can be calculated. A similarity matrix is used to represent the difference between the number of times that SU A was wrongly estimated as SU B and the number of times that SU B was wrongly estimated as SU A. To constrain the results into a fixed space, the resulting difference is divided by the sum of the two conditions. This is done using the equation

$$\boldsymbol{S} = \frac{\left|\boldsymbol{R}_{\boldsymbol{hist}}^{\boldsymbol{T}} - \boldsymbol{R}_{\boldsymbol{hist}}\right|}{\boldsymbol{R}_{\boldsymbol{hist}}^{\boldsymbol{T}} - \boldsymbol{R}_{\boldsymbol{hist}}} \tag{20}$$

where S is the similarity matrix, R_{hist} is the matrix of histogram results and $(\quad)^T$ is the matrix transpose. For this equation, the division is a point-wise division and R_{hist} is a square matrix.

The resulting similarity values are shown in Figure 31. The values for the given radios, since they lie along the diagonal, remain the same in the transpose and therefore have a difference value of zero. It can also be seen that radios using the same channel selection strategies, which are radios 11 & 12, 13 & 14, 15 & 16, and 17 & 18, have a very low difference value for the corresponding radio. However, there are several values which are also low-valued which do not share the same channel selection strategy. One example is the value set associated to the relationship between SUs 15 and 18. These two radios do not share the same channel selection method, and such a result increases the probabilities of false positives. Therefore a method of determining which of these are statistically insignificant enough to ignore is desired.

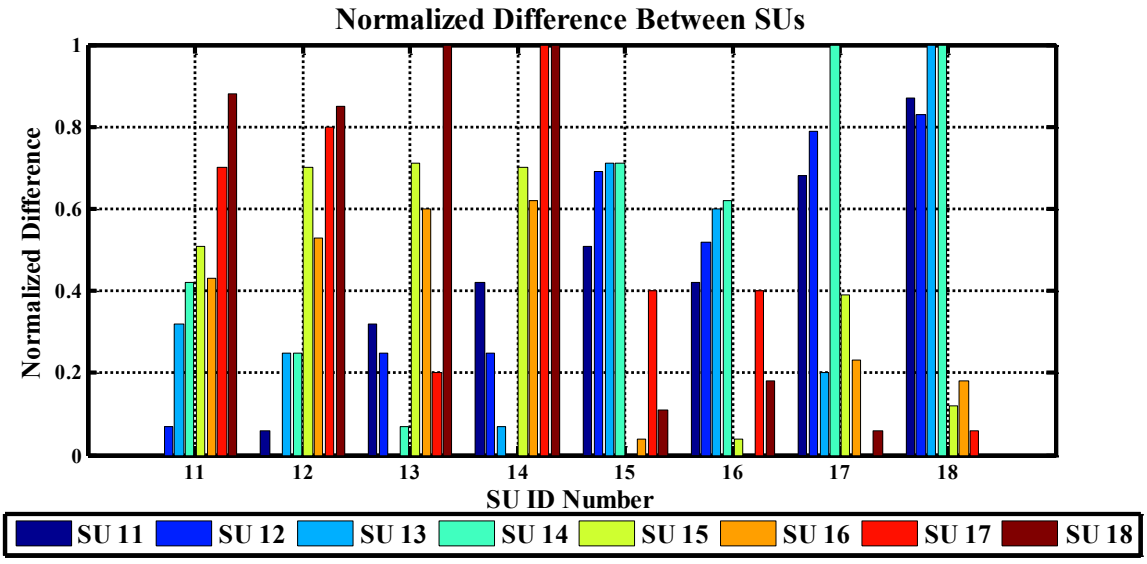

Figure 31: Normalized difference between the SUs over 5,000 channel decisions.

6.3.3 Removing False Groupings

One robust method used to detect and remove these values is to discard counts that are below the mean within each count matrix, not including values on the diagonal. This is computed for each dimension, and is discarded if it is not above the mean for both. For

example, for Figure 30, this threshold would be set at 35 for SU 18, and for Figure 29, this threshold would be set at 26 for SU 18 in. The remaining similarity values can be seen in Figure 32. In this case, the number of removed cases was quite large, and leaves a minimum of values behind. This step will become more critical as the number of training and testing steps is decreased. Even here, it can be seen that the possible false detection for SU 18 in the SU 15 group has been eliminated.

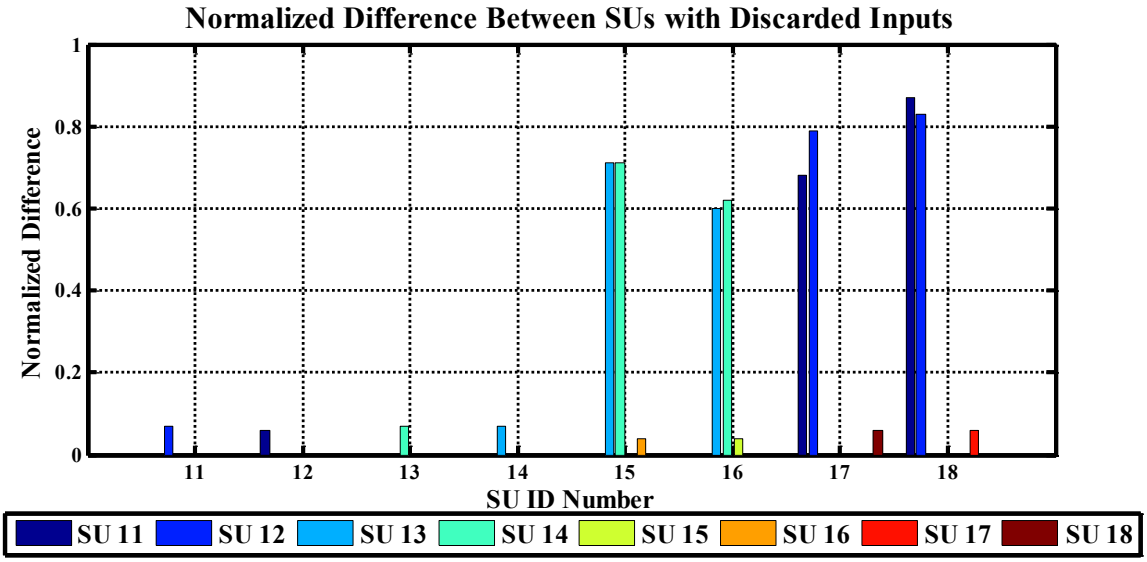

Figure 32: Normalized difference between the SUs with discarded inputs over 5,000 channel decisions.

Decreasing the number of training inputs to the histogram method drastically decreases the effectiveness of the process. Figure 33 shows the similarity values for a randomly selected section of a simulation taken over 200 channel selections. As the figure shows, this method does not handle the reduction in training sequences well.

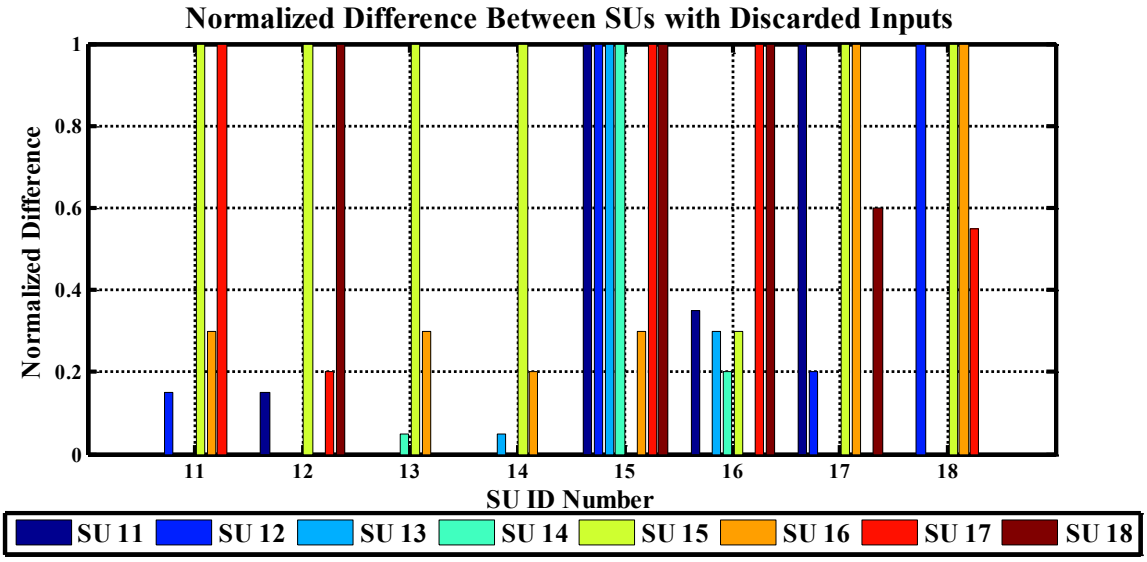

Figure 33: Normalized difference between the SUs as calculated by the classifier with discarded inputs over 200 channel decisions.

Therefore, while this would be useful for radios using slowly changing channel selection methods, this is not one that would be effective for systems which use change channel selection algorithms more readily, or for short observation windows. The following section introduces a second method of determining the channel selection method with a much shorter training requirement.

6.4 SU Grouping: Bayesian Probability

Rather than using the most probable state of the classifier as a discrete input, a more accurate and faster approach to this problem would be to use the probability estimates that are calculated within the classifier itself. This gives an even better understanding of the relationships between different SUs. This method will also reduce the number of observed channel selections by a SU that is needed for the observer and its classifier to be able to make accurate grouping decisions.

6.4.1 Classification Method

To train and test the classifier, the backward classification method described in Section 6.2.2. For the first tests, we employ the same simulation profile we used for the histogram testing. This simulation profile used 10 PUs and 8 SUs, where four channel selection methods are used, with two SUs using each selection method.

However, there is a slight deviation from the previous section's terminology in this section. Previously, the phrase testing length has referred to the number of channel selection instances used as inputs to the classifier. But for this section, the testing length will refer to the number of classifier outputs, which are the probabilities of occurrences, which are to be used as inputs to the channel selection estimator.

While the number of tested channel section instances is reduced using this method, observing multiple channel decisions by a SU before calculating its channel selection method is still required to help smooth the statistics. This is illustrated in Figure 34, which shows a plot of the probability of occurrence as output by the classifier for SU 13 and 14 for channel decisions made by SU 13, as determined by the truth. This shows how the user probabilities evolve over time, as well as potential incorrect SU ID classifications. This occurs where other SUs, such as SU 14, have a higher probability than SU 13. Only a small portion of time is plotted to better show detail.

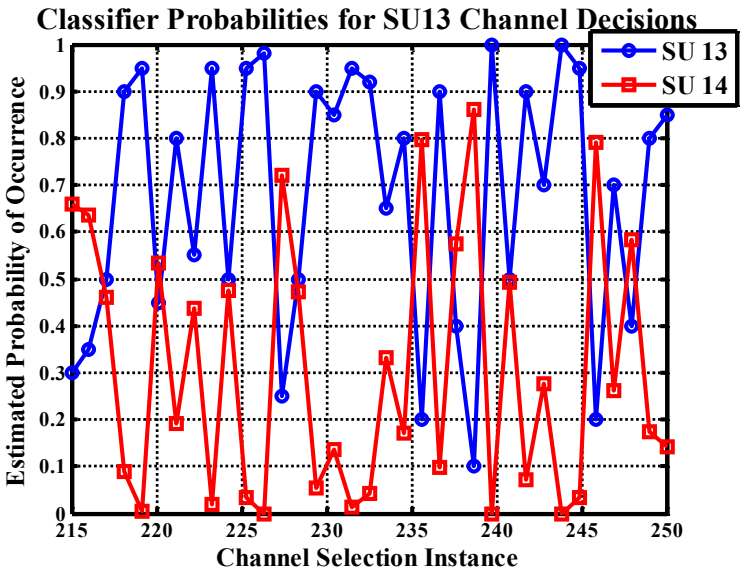

Figure 34: Classifier probability outputs for SU 13 and 14 for channel decisions made by SU 13 as a function of time.

The figure shows that there can be a considerable amount of volatility in the classifier probabilities between multiple channel selection instances. However, if looked at in a broader sense, the average of these values can be very useful. The figure shows that SU 13 maintains a very high probability of occurrence throughout the plot. However, SU 14, which shares the same channel selection strategy, often has a probability of occurrence that is much higher than zero. All other possible SUs have a near-zero probability of occurrence. Since the summation of all probabilities of occurrence must equal one, one can see that the other SUs have negligible probabilities since the summation of SU 13 and SU 14 equal nearly one. The sum of all other SU probabilities never equal more than 0.05 for this simulation.

6.4.2 Detection Threshold Calculation

The most direct method for using this as a grouping method is the determination of a threshold value. Before a threshold can be determined, the effect of changing the lengths of training inputs needs to be analyzed.

Therefore multiple, independent simulations are used to develop a reliable probability model. This is done using a histogram of the output probabilities over each full simulation for each SU. From this data, the cumulative distribution function (CDF) corresponding to the probabilities returned by the classifier can be determined. This CDF represents the probability of a missed detection for any given decision threshold.

Figure 35 shows the CDF for a single sample test. The red line, which is lowest on the plot, corresponds to the tested radio, SU 13, and the light blue, second lowest, line represents SU 14, which shares the same channel selection method. From this, the probability that a similarity would be detected can be calculated for a given probability threshold. This illustrates the probability that, for a single SU 13 channel decision, the probability of occurrence from the classifier for each SU is below the threshold value. For SUs that do not share the same channel selection method, this represents one minus the probability of false alarm.

For example, if a detection threshold of 0.5 is used, then the similarity between the two radios would have a missed detection rate of approximately 73 percent, and an accompanying probability of detection of approximately 27 percent. The probability of detection for the other radios can be calculated using a similar process.

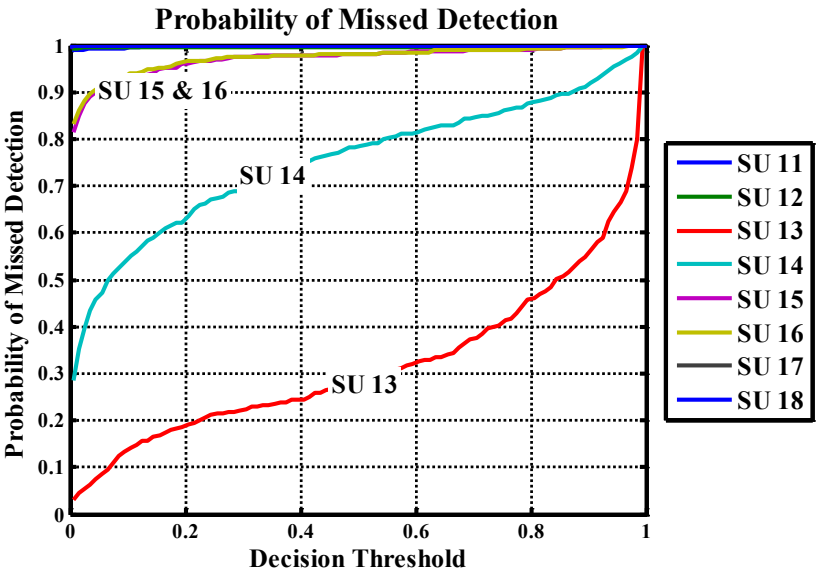

Figure 35: Probability of missed group detection for a range of threshold values using a single classifier output from an SU 13 channel selection.

This method is clearly suboptimal, and can be improved by increasing the number of data points. The next step is to determine the probability of missed detection for a range of test lengths where the probabilities from the classifier are averaged over a fixed length channel selections. This allows us to determine the number of observed channel selection instances for each SU needed to achieve the desired detection rate when averaging the probability of occurrence.

The same dataset used to calculate the probability of missed detection curves in Figure 35 are used here. For this test, the probability of missed detection is calculated for SU 14 when a channel is selected by SU 13. The results are shown in Figure 36, where the numbers listed in the legend represent the number of channel decisions by SU 13 the probability of occurrence was averaged over. The blue line is the same curve found in Figure 35 for SU 14.

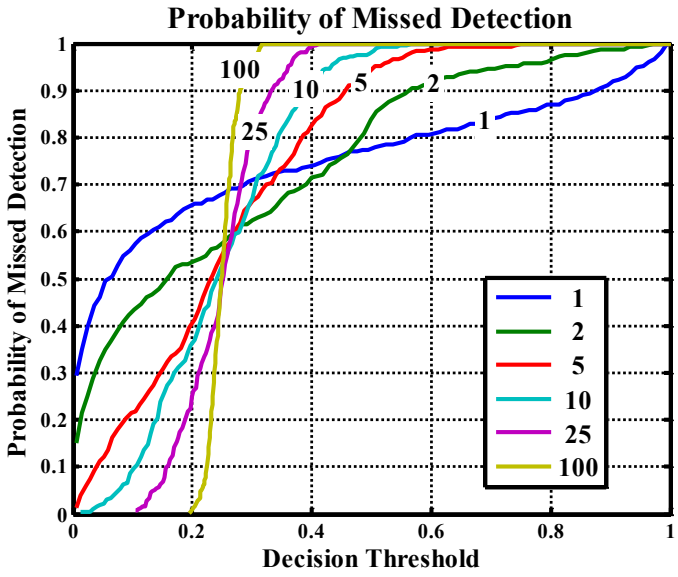

Figure 36: Probability of missed group detection for SU 14 for a range of threshold when averaging over multiple channel selections by SU 13.

Figure 37 shows the effects of the summation over a series of inputs on an incorrect selection, in this case SU 16, again for channel selection instances for SU 13. SU 16 was chosen because it had the highest probability of detection at higher threshold levels. Increasing the length over which the means are calculated, it can be seen that it decreases the probability that the SU will be incorrectly assigned as a member of that group. The x-range of the plot is limited to 0 to 0.5 to highlight the low-threshold performance.

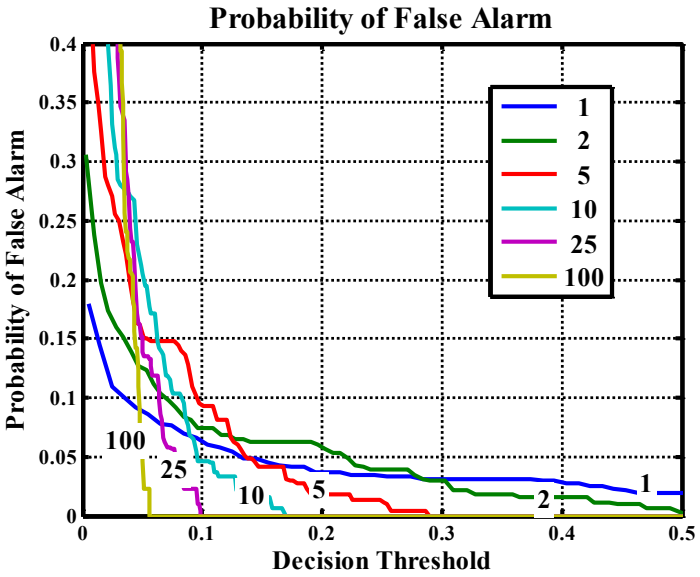

Figure 37: Probability of a false group detection.

While increasing the number of samples that are averaged prior to threshold testing may increase certainty, it has a time cost associated with it. One reason this method was investigated was to facilitate faster selection of the channel selection method groups. By increasing the number of channel selections which are averaged, this effectively removes that property. Also, increasing the averaging length increases the amount of time that is required for the system to detect if the SU changes its channel selection method.

For the 200 channel decision input length used in the standard simulation profile, each SU makes, on average, 25 decisions. Therefore, to compare similar performance between this method and the histogram method described in, the comparison is done using an averaging length of 25. In a theoretical analysis using the results in Figure 36 and Figure 37, the probability of detection can be calculated.

From these we choose a threshold value of 0.15. For this threshold value, it is expected that the similarity between SU 13 and SU 14 would be detected approximately 95

percent of the time. It is also expected that SU 16 and SU 17 would each be falsely detected as using the same channel selection technique as SU 13 less than 0.05 percent of the time.

6.4.3 Two SUs Per Selection Method Results

Now that a theoretical result has been calculated, a simulation is run to confirm the theoretical results. This simulation has the same parameters that were used to model the theoretical results, with testing conducted in two phases. The first is testing SU 13 specifically, to confirm the results specific to it in the theoretical section. The next analyzes the performance of the entire network.

The first test of SU 13 is a straight-forward test, simply testing the detection rates for all radios given a set threshold. From the theoretical tests, a good starting threshold seems to be about 0.15, given that the inputs are averaged over a length of 25 channel decisions by SU 13. Figure 38 shows the grouping results of the simulation. The accuracy prediction for SU 14 was extremely close, with the simulation results showing a 94 percent detection rate. The results for SU 15 had exceeded the threshold more often than predicted, but SU 16 did not get detected at all. However, SUs 11, 12, 17, and 18 performed as expected. The switch between SU 15 and 16 is not unexpected, since they use the same channel selection technique, and the percent of the time it was incorrectly associated meets the theoretical expectation for SU 16.

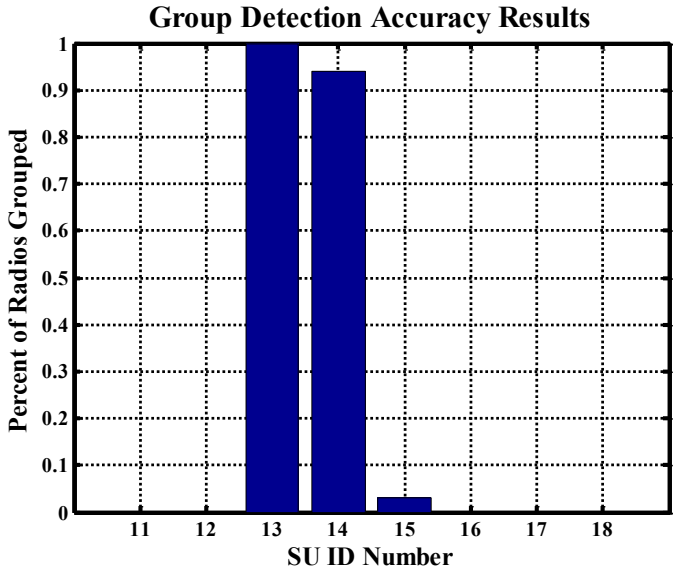

Figure 38: Simulation results for group detection.

The next step is to test the use of the threshold for all radios. A new simulation is conducted using a threshold of 0.15 and an averaging length of 25. This simulation uses the same properties as the previous eight-SU simulations. Figure 39 shows the results of the simulation. To make reading the plot easier, incorrect groupings were made negative in value, correct groupings left positive, and the SU for that group is set to a value of one. A numerical analysis of the simulation can be found in Table 11.

This highlights the grouping differences between the different channel selection techniques, where least recently occupied (11 & 12) and least frequently accessed (13 & 14) have the best performance, and least occupied (14 & 15) has the worst. These results are particularly interesting when comparing the incorrect groupings. One example is SUs 13 and 14 are often grouped with SU 15 and 16, but this is not true in reverse. This is because the least occupied channels are likely to be infrequently accessed, but infrequently accessed channels might not be the least occupied.

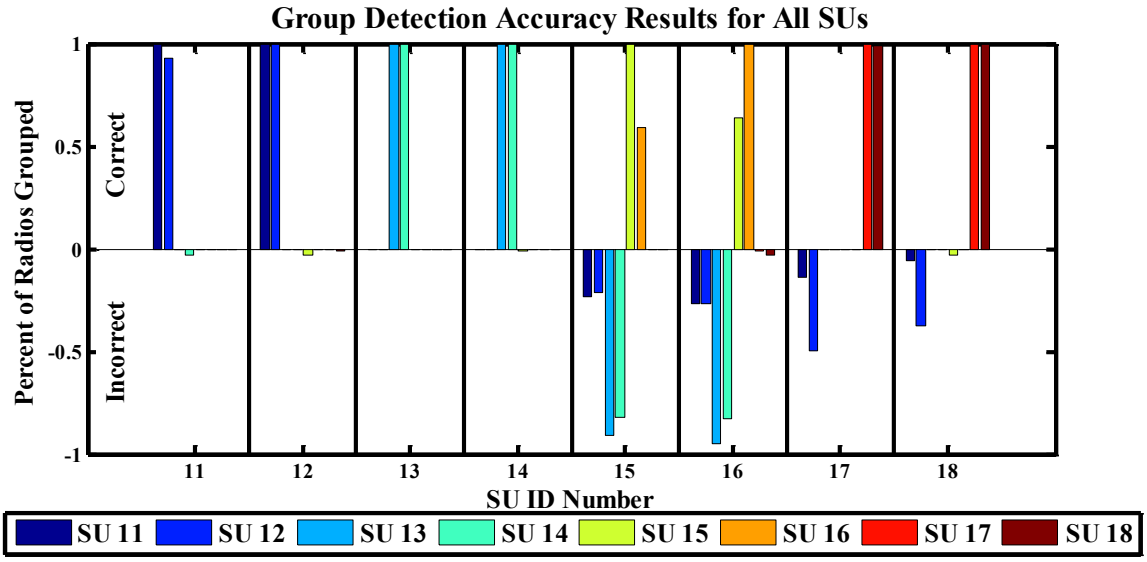

Figure 39: Channel group detection for two SUs per channel selection method.

Overall grouping accuracy (number of correct groupings/total possible)	82 %
False grouping rate (num. incorrect / num. grouped)	27 %

Table 11: Statistics for First SU Grouping Test

Using the duality in the groupings between radios greatly decreases the false alarm rate. This is implemented by assuming that if two radios are using the same channel selection method, there will be near equal levels of confusion between the two. For example, radio pairs using the same channel selection method can be seen to have similar levels of confusion. However, incorrect classifications, such as SU 15 incorrectly grouping SU 13 with itself, do not have an equal level of confusion with SU 13 grouping SU 15 with itself, making this likely incorrect. Using this, the false alarm rate drops from 27 to less than one percent.

6.4.4 Different Numbers of SUs Per Selection Method Results

To test this, only three channel selection methods are used, but each is used by a different number of radios. The simulation profile for the SUs is shown in Table 12. The

only thing to change compared to the SU properties shown in Table 10 is a different number and distribution of the channel selection methods.

	Channel Selection Index	Activity Lambda	Idle Lambda
SU 1 (ID 11)	3	10	30
SU 2 (ID 12)	3	10	30
SU 3 (ID 13)	4	10	30
SU 4 (ID 14)	6	10	30
SU 5 (ID 15)	3	10	30
SU 6 (ID 16)	3	10	30
SU 7 (ID 17)	6	10	30
SU 8 (ID 18)	6	10	30

Table 12: SU Properties for Final Simulation Testing

The results of the SU grouping rates can be found in Figure 40. As with the previous set of results for the SU grouping rates, all incorrect grouping decisions are shown as negative-valued, and all correct grouping decisions are positive. Table 13 contains the calculated accuracies for this simulation.

This shows a significant drop in the grouping accuracy, but some drop in the accuracy in expected. This simulation only used channel selection indices three, four, and six, where channel selection index six is used by three SUs. From the previous section, it is shown that channel selection index 6, least occupied, performs poorly. This method was included for this test to demonstrate a floor in performance. If least occupied is replaced with channel selection index five, least frequently accessed, the grouping performance would increase.

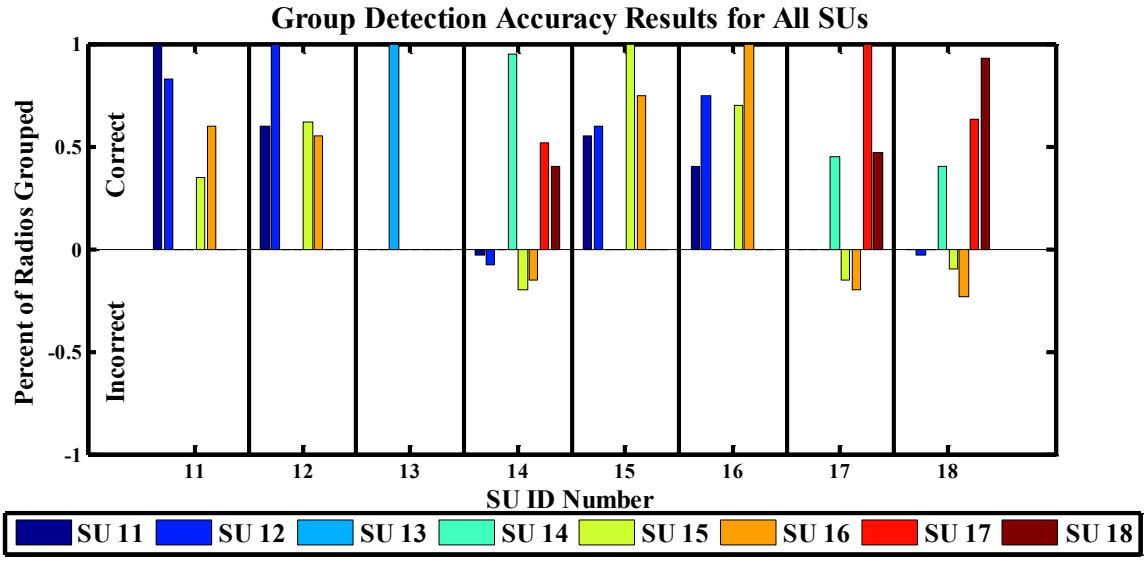

Figure 40: Group detection for more than two SUs per channel selection method.

Overall grouping accuracy (number of correct groupings/total possible)	58 %
False grouping rate (num. incorrect / num. grouped)	7 %

Table 13: Statistics for Second SU Grouping Test

The result for the grouping accuracy is still lower than expected when using the same threshold value. Plotting the probability of missed detection curves in Figure 41 shows that the inclusion of more than one other radio using the same channel selection method requires a lower threshold value. This is calculated in the same manner as the previous probability of missed detection plots. However, for this test the channel decisions made by SU 15 are used instead of SU 13, which was used in Figure 35. The figure shows that to maintain the same performance for group detection, a threshold of approximately 0.10 is needed. However, lowering the threshold increases the false detection probability slightly.

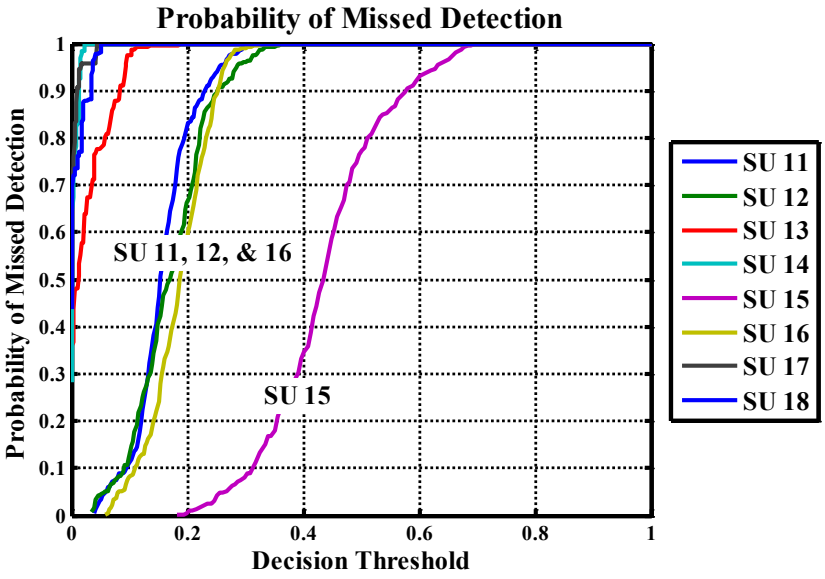

Figure 41: Probability of missed detection for the new simulation case.

The re-running the grouping method on the simulation results with the new threshold value returns the grouping results found in Figure 42. This shows an increase in most of the correct grouping columns. But the additional grouping detection comes at the cost a higher false detection rate. The statistics for the calculation can be found in Table 14.

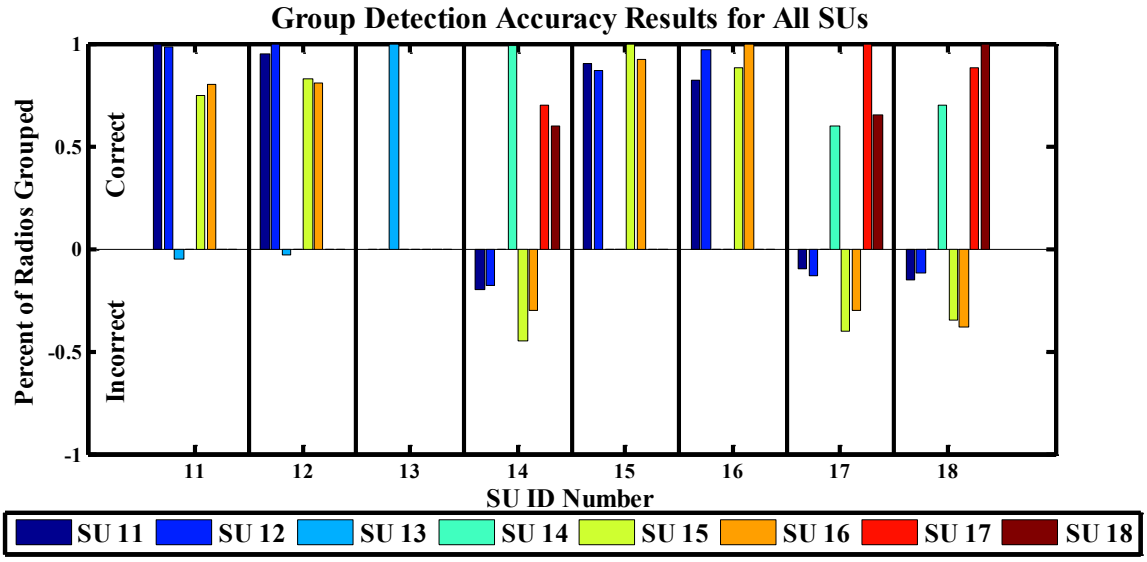

Figure 42: Group detection for more than two SUs per channel selection method with a lowered threshold value.

Overall grouping accuracy (number of correct groupings/total possible)	72 %
False grouping rate (num. incorrect / num. grouped)	16 %

Table 14: Statistics for Second SU Grouping Test with Lowered Threshold Value

With the reduction in the threshold value, the number of missed detections of the proper grouping values dramatically decreased. However, this also increased the number of false positives. These are the tradeoffs which need to be made in any system implementing this method. However, the overall accuracy result is expected.

In Figure 41, the SUs that use the same channel selection method as SU 15 have an average error of 9 percent each. Therefore, between the three of them, a loss in accuracy of approximately 18 percent is expected. This theoretical result is confirmed by the simulation test. And comparing Figure 39 and Figure 42, it can be seen that Figure 42 has a much lower false alarm rate, due to least frequently accessed not being used, and therefore incurring a much lower number of false alarms.

7

Conclusion

This thesis introduces the UDNS, a DSA network simulator that allows the modeling of arbitrarily large DSA networks. While it currently does not include all of the complex channel selection and data transmission methods that would be encountered in a real-world environment, these functions can be easily added. The UDNS is also an effective tool in exploring low-cost estimation of secondary user radio properties.

The largest contribution is the investigation of these estimations when using energy detection, one of the most basic channel sensing methods. Energy detection requires only basic RF sensing equipment, and can be implemented cheaply and efficiently in mobile hardware. However, this method of channel sensing only provides a minimum of information to the mobile unit. By combining energy detection with the computationally efficient Naïve Bayesian classifier, we can use this simplistic sensing method to provide a user or observer an incredible amount knowledge about the other users in a DSA network.

Energy detection has the added benefit that it does not require any knowledge of the messages being transmitted. This allows it to work, not only on resource-constrained handhelds, but on networks were unknown data encoding would render a more direct approach useless. This also allows a handheld to monitor many more channels than if it were

using more complex methods. This greatly increases the number of channels that it can utilize, increasing its flexibility, and increasing the overall channel utilization.

7.1 Future Work

Future work in this area should focus on increasing the realism of the simulations used for the classification testing. The first change to explore is the relocation of the secondary users and the observer node. This changes the noise and signal power readings for each, and more accurately simulates the real-world.

The next logical step would be to include more complex channel selection techniques. The methods implemented currently in the UDNS are not complex, but are integral parts of the most common, more complex channel selection techniques.

Finally, the modulation schemes that are used by each user would be changed. This the least important factor to test since this likely will have the least impact on the classification accuracy. This is due to the channel sensing technique used, since it is waveform agnostic.

We hope that this research will serve as a starting point for further research on resource-constrained DSA networks. DSA networks show promise as a method of increasing the utilization of the existing spectrum without requiring major changes to the FCC's licensing.

References

[1] MacKenzie, A.B.; Reed, J.H.; Athanas, P.; Bostian, C.W.; Buehrer, R.M.; DaSilva, L.A.; Ellingson, S.W.; Hou, Y.T.; Hsiao, M.; Jung-Min Park; Patterson, C.; Raman, S.; da Silva, C.; , "Cognitive Radio and Networking Research at Virginia Tech," *Proceedings of the IEEE* , vol.97, no.4, pp.660-688, April 2009

[2] lleri, O.; Mandayam, N.B.; , "Dynamic spectrum access models: toward an engineering perspective in the spectrum debate," *Communications Magazine, IEEE* , vol.46, no.1, pp.153-160, January 2008

[3] "Electronic Code of Federal Regulations." *GPO Home Page*. Web. 17 July 2011. <http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr>.

[4] Bennai, M.; Sydor, J.; Rahman, M.; , "Automatic channel selection for cognitive radio systems," *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on* , vol., no., pp.1831-1835, 26-30 Sept. 2010

[5] Buddhikot, M.M.; Ryan, K.; , "Spectrum management in coordinated dynamic spectrum access based cellular networks," *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on* , vol., no., pp.299-307, 8-11 Nov. 2005

[6] Joseph Mitola III, "Software radios: Survey, critical evaluation and future directions," IEEE Aerospace and Electronic System Magazine, vol. 8, Issue. 4, pp. 25-36, April 1993.

[7] Mitola, J. Software Radio Architecture. John Wiley and Sons, 2000.

[8] Mitola, J. Cognitive radio: An integrated agent architecture for software defined radio. Ph.D. Dissertation, KTH, 2000.

[9] Mitola, J., III; Maguire, G.Q., Jr.; , "Cognitive radio: making software radios more personal," *Personal Communications, IEEE* , vol.6, no.4, pp.13-18, Aug 1999

[10] Clancy, Thomas C. "Dynamic Spectrum Access in Cognitive Radio Networks." Diss. University of Maryland, 2006. Web. 1 July 2011. <www.cs.umd.edu/~jkatz/THESES/clancy.pdf>.

[11] Changwoo Lee; Wonjun Lee; , "Exploiting Spectrum Usage Patterns for Efficient Spectrum Management in Cognitive Radio Networks," *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on* , vol., no., pp.320-327, 20-23 April 2010

[12] D. Niyato and E. Hossain, "Cognitive radio for next-generation wireless networks: An approach to opportunistic channel selection in IEEE 802.11-based wireless mesh," IEEE Wireless Communications, vol. 16, no. 1, pp. 1536–1284, February 2009.

[13] Min, A.W.; Kyu-Han Kim; Shin, K.G.; , "Robust cooperative sensing via state estimation in cognitive radio networks," *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on* , vol., no., pp.185-196, 3-6 May 2011

[14] Wenbo Wang; Kwasinski, A.; , "Feedback-based cooperative primary channel activity estimation for dynamic spectrum access," *Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on* , vol., no., pp.1-5, 7-10 Nov. 2010

[15] Jian Chen; Gibson, A.; Zafar, J.; , "Cyclostationary spectrum detection in cognitive radios," *Cognitive Radio and Software Defined Radios: Technologies and Techniques, 2008 IET Seminar on* , vol., no., pp.1-5, 18-18 Sept. 2008

[16] Sutton, P.D.; Ozgul, B.; Nolan, K.E.; Doyle, L.E.; , "Bandwidth-Adaptive Waveforms for Dynamic Spectrum Access Networks," *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on* , vol., no., pp.1-7, 14-17 Oct. 2008

[17] R. Cagley, S. McNally, and M. Wiatt, "Dynamic channel allocation for dynamic spectrum use in wireless sensor networks," in IEEE Military Communications Conference. IEEE, October 2006, pp. 1–5.

[18] Datla, D.; Wyglinski, A.M.; Minden, G.J.; , "A Spectrum Surveying Framework for Dynamic Spectrum Access Networks," *Vehicular Technology, IEEE Transactions on* , vol.58, no.8, pp.4158-4168, Oct. 2009

[19] K. Tsukamoto, Y. Omori, O. Altintas, M. Tsuru, and Y. Oie, “On spatially-aware channel selection in dynamic spectrum access multi-hop inter-vehicle communications,” 70th IEEE Vehicular Technology Conference, pp. 1–7, September 2009.

[20] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “A survey on spectrum management in cognitive radio networks,” IEEE Communications Magazine, vol. 46, no. 4, pp. 40–48, April 2008.

[21] Watanabe, K.; Ishibashi, K.; Kohno, R.; , "Performance of Cognitive Radio Technologies in the Presence of Primary Radio Systems," *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on* , vol., no., pp.1-5, 3-7 Sept. 2007

[22] Guangxiang Yuan; Yi Zhu; Yu Zhang; Yang Yang; Wenbo Wang; Zhong Fan; Sooriyabandara, M.; , "Multi-User Cooperation for Channel Selection in Cognitive Radio Networks: A Bayesian Approach," *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference* , vol., no., pp.1-5, 6-10 Dec. 2010

[23] Baldo, N.; Tamma, B.R.; Manojt, B.S.; Rao, R.; Zorzi, M.; , "A Neural Network Based Cognitive Controller for Dynamic Channel Selection," *Communications, 2009. ICC '09. IEEE International Conference on* , vol., no., pp.1-5, 14-18 June 2009

[24] Beibei Wang; Zhu Ji; Liu, K.J.R.; , "Primary-Prioritized Markov Approach for Dynamic Spectrum Access," *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on* , vol., no., pp.507-515, 17-20 April 2007

[25] Butun, I.; Cagatay Talay, A.; Turgay Altilar, D.; Khalid, M.; Sankar, R.; , "Impact of mobility prediction on the performance of Cognitive Radio networks," *Wireless Telecommunications Symposium (WTS), 2010* , vol., no., pp.1-5, 21-23 April 2010

[26] Kaneko, S.; Nomoto, S.; Ueda, T.; Nomura, S.; Takeuchi, K.; , "Predicting Radio Resource Availability in Cognitive Radio - an Experimental Examination," *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on* , vol., no., pp.1-6, 15-17 May 2008

[27] Wang Ding; Songnian Yu; Qianfeng Wang; Jiaqi Yu; Qiang Guo; , "A Novel Naive Bayesian Text Classifier," *Information Processing (ISIP), 2008 International Symposiums on* , vol., no., pp.78-82, 23-25 May 2008

[28] Kelemen, A.; Hong Zhou; Lawhead, P.; Yulan Liang; , "Naive Bayesian classifier for microarray data," *Neural Networks, 2003. Proceedings of the International Joint Conference on* , vol.3, no., pp. 1769- 1773 vol.3, 20-24 July 2003

[29] B. Hilburn, T. R. Newman, T. Bose, and S. L. Kadambe, “A survey of basic channel selection techniques for cognitive radios,” Proc. of the SDR Wireless Innovation Conference, pp. 37-41, Nov.-Dec. 2010.

[30] D. Cabric, A. Tkachenko, and R. W. Brodersen, “Experimental study of spectrum sensing based on energy detection and network cooperation,” Proc. of the 1st International Workshop on Technology and Policy for Accessing Spectrum, vol. 222, no. 12. ACM, 2006.

[31] Saeed, R.; , "Cognitive Radio and advanced spectrum management," *Communications, Computers and Applications, 2008. MIC-CCA 2008. Mosharaka International Conference on* , vol., no., pp.xii, 8-10 Aug. 2008

[32] Cuiran Li; Chengshu Li; , "Opportunistic spectrum access in cognitive radio networks," *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on* , vol., no., pp.3412-3415, 1-8 June 2008

[33] Zhe Chen; Nan Guo; Qiu, R.C.; , "Building a cognitive radio network testbed," *Southeastcon, 2011 Proceedings of IEEE* , vol., no., pp.91-96, 17-20 March 2011

[34] Pal, R.; Idris, D.; Pasari, K.; Prasad, N.; , "Characterizing reliability in cognitive radio networks," *Applied Sciences on Biomedical and Communication Technologies, 2008. ISABEL '08. First International Symposium on* , vol., no., pp.1-6, 25-28 Oct. 2008

[35] Xiangwei Zhou; Ye Li; Young Hoon Kwon; Soong, A.; , "Detection Timing and Channel Selection for Periodic Spectrum Sensing in Cognitive Radio," *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE* , vol., no., pp.1-5, Nov. 30 2008-Dec. 4 2008

[36] Xiangquan Zheng; Ying Li; Haicheng Zhang; , "A collision-free resident channel selection based solution for deafness problem in the cognitive radio networks," *Wireless Information Technology and Systems (ICWITS), 2010 IEEE International Conference on* , vol., no., pp.1-4, Aug. 28 2010-Sept. 3 2010

[37] Zhao Bing; Guo Lili; , "Research of Spectrum Detection Technology in Cognitive Radio," *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on* , vol.1, no., pp.188-191, 25-26 April 2009

[38] Mitchell, Tom M. "Generative and Discriminative Classifiers: Naïve Bayes and Logistic Regression." *Machine Learning*. McGraw Hill, 2010. 1-17. 19 Jan. 2010. Web. 12 July 2011. <www.cs.cmu.edu/~tom/mlbook.html>.