Machine Learning based Spectrum Sensing in Cognitive Radio Networks

by

Raunaq Singh Jabbal

BCS 063

A report submitted for Summer Project

Bachelor of Technology

in

CSE

ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Report Certificate

I hereby certify that the work, which is being presented in the report, entitled **Machine Learning based Spectrum Sensing in Cognitive Radio Networks**, for Summer Project in Computer Science Engineering and submitted to the institution is an authentic record of my own work carried out during the period *May-2022* to *August-2022* under the supervision of **Dr. Binod Prasad**. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Name and Signature of the candidate

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: Name and Signature of the Research Supervisors

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the proper format as specified in the guidelines for thesis preparation.
I declare that the work containing in this report is my own work. I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name:

Roll. No:

Date:

Abstract

The number of wireless communication devices has grown exponentially, and with the advent of 5G, demand for spectrum bands is increasing. We use Cognitive Radio to bridge the gap between supply and demand. Spectrum Sharing is a concept where other devices can access the spectrum band belonging to licensed users who are not using it. Cognitive Radio is a wireless communication mechanism to intelligently detect which channels are in use and reuse the channel where the PU is not transmitting. Spectrum Sensing is the critical part of Cognitive Radio, where we sense the spectrum band periodically, which allows us to sense holes in the spectrum. Secondary Users (SU) sense the spectrum for the presence of Primary Users (PU). They can use various algorithms to determine the presence or absence of primary users and can use that spectrum band when the licensed user is not transmitting. There are various factors on which the decision depends, the number of SUs, the number of PUs, the distance between SUs and PUs, noise levels, SNR values, type of fading used, sensing time, and type of technique used to determine the outcome. Techniques include Non-cooperative Spectrum Sensing, where individual SUs determine independently, and cooperative spectrum sensing (CSS), where all SUs collectively decide if the PU is active or not, because at one moment, a particular SU may report incorrectly. Under cooperative sensing, various conventional techniques like AND-based, OR-based, and Maximum Ratio Combining (MRC) based.

Machine Learning is a field of study based on building models and methods that "learn," which use existing data to come up with a solution and increase their performance to predict outcomes or decisions accurately. Machine learning is closely related to Artificial Intelligence and Data Science. Deep Learniing is a subset of Machine Learning which tries to mimic the human brain to make decisions based on given data, and this approach gives them some advantages over Machine Learning.

This paper summarizes effects of parameters like the distance between PU and SUs, number of SUs, sensing time, fading scenarios, multiple cooperative spectrum sensing algorithms, classical and Machine Learning techniques, and Non-cooperative Spectrum Sensing algorithms.

Dedication

I am dedicating this project to my family and my friends. My parents and my brother, whose words of encouragement ring in my ears. Their love for me knows no bounds. I will alwyas appreciate all they have done like developing my skills, teaching me the value of hard work, always providing constructive criticism, and for being a constant source of support and guidance. Also, I dedicate this thesis to all friends who always constantly aided me whenever I required help.

Acknowledgments

I would like to express my special thanks of gratitude to my mentor Dr. Binod Prasad for giving me this opputunity to do this wonderful Summer Project on the topic Machine Learning based Spectrum Sensing in Cognitive Radio Networks. It was a great honor and privilege to study and learn under his guidance.

I am extremely grateful to my parents and my brother for their sacrificies, love and for mentally preparing and supporting me throughout this project. They have always been a source of inspiration and their assistance has gone a long way for me and the completion of this project.

My completion of this project could not have been accomplished without the support of my friends and classsmates, this project is a success because of their endless help and support.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Context

There has been an increase in wirelessly communicating devices because of the development of wireless technologies and protocols. This development has led to an increase in demand for radio spectrum. Cognitive Radio is a concept that allows SUs to use licensed spectrum belonging to Primary Users to overcome this demand increase. Spectrum Sensing is the part of Cognitive Radio where the Secondary Users listen to the spectrum band and use the sensing data to determine if the Primary User is transmitting. Spectrum Sensing is of two types, non-cooperative and cooperative. Non-cooperative Spectrum Sensing is where an SU independently decides if the PU is active. Cooperative Spectrum Sensing is where multiple SUs use an algorithm to decide if the PU is active. Cooperative Spectrum Sensing is more accurate than Non-cooperative Spectrum Sensing because that one SU may not sense appropriately because of the environment, so its decision may be incorrect. When multiple SUs work together, they can overcome this issue. Then we may use a variety of algorithms that may be good at spotting outliers, leading to an improvement in the decision. In Non-cooperative spectrum sensing [5], the SU senses the energy it receives, and only if it is below a certain threshold does the SU conclude that the PU is not actively using the licensed spectrum band, and it can reuse that same band. Non-cooperative spectrum sensing is of two types, classical techniques, and Machine Learning techniques.

Classical techniques [21] are based upon Cooperative Spectrum Sensing, where multiple SUs aid in deciding the final outcome, which consists of AND, OR, and Maximum Ratio Combining

techniques. In AND technique, only when all SUs conclude that the PU is transmitting, the final decision is that the PU is transmitting and the SUs should not transmit. In the OR technique, even if one SU concludes that the PU is actively transmitting, the final decision is that the PU is transmitting. In Maximum Ratio Combining, sensed energy value of each SU of multiplied by the normalised average SNR value, and the summation of these new energy values is compared against the threshold value to give the final decision. This is done so that the energy value of SUs having high SNR values are given more importance, but this requires us to know the SNR value.

Machine Learning algorithms are handy in a wide range of fields. These algorithms can automatically understand and extract patterns from data and apply this understanding to new data. We can classify our problem into two types, classification and regression. Classification is where the output type is a discrete set of values, like a Yes/No decision. Regression is where the output type is a continuous set of values.

Further, Machine Learning algorithms can be classified into Supervised Learning, Unsupervised Learning, and Reinforcement Learning. The dataset the algorithm receives is labeled with output values in Supervised Learning, and in Unsupervised Learning, there is no labeled dataset. In Reinforcement Learning, the algorithm is rewarded for doing a desirable behavior and punished for doing an undesirable behavior, and the algorithm learns through trial and error. Another crucial part of Machine Learning is Neural Networks, which try to mimic brain activity to come to a decision, are very flexible, and the algorithms can be applied to various problems. For Machine Learning approach, we do not use Non-cooperative Spectrum Sensing approach of choosing a threshold, like we do in Classical Algorithms. Instead, we pass the energy values of all SUs to the algorithms as dataset, and the algorithm works on those values to conclude if the PU is transmitting.

Spectrum Sensing is a Classification problem because we need to conclude if the PU is actively transmitting or not, so the algorithm will only give out two values, 0 and 1, representing a Yes and No. We have chosen various Supervised Learning algorithms Logistic Regression, Linear Support Vector Machine, Gaussian Support Vector Machine, K Nearest Neighbours, Random Forest Classification, Naïve Bayes, Artificial Neural Networks, and Gradient Boosting Libraries like

CatBoost, XGBoost, and ADABoost.

1.2 Problem/Motivation

We require detailed results of the effect of various parameters on the Spectrum Sensing part of Cognitive Radio. Along with this, we need to know performance of Spectum Sensing under various fading scenarios and algorithms.

The goal of spectrum sensing is to decide between the two hypotheses [21]:

$$z(t) = \begin{cases} n(t) & H_0 \text{ (white space)} \\ hs(t) + n(t) & H_1 \text{ (occupied)} \end{cases} \tag{1.1}$$

Where z(t) is the signal sample received by the SU, s(t) is the transmitted signal of the primary user, n(t) is the Additive White Gaussian Noise (AWGN), and h is the complex gain of the channel.

1.3 Objectives

Spectrum Sensing is the most important part of Cognitive Radio where SUs listen, or sense the spectrum and based on the sensed data decide if the PU is using the spectrum band or not. This paper thoroughly reviews and considers various aspects like algorithms, sensing parameters, fading scenarios stated below

- Compare performance between:

  Non-cooperative Spectrum Sensing technique, Classical Cooperative Spectrum Sensing Techniques (AND, OR and MRC) and various Machine Learning Algorithms in Cooperative Spectrum Sensing.

- To show the effect of Rayleigh Fading, Rician Fading, Nakagami Fading, and AWGN at Variance values 1 and 2 on Spectrum Sensing.

- Show effect of Sample Size or Sensing Time, SU numbers, Training Dataset Size for Machine Learning Algorithms on Spectrum Sensing.

  The main performance metric is the Receiver Operating Characteristic Curve (ROC Curve) and Area under the ROC Curve (AUC value). The code enables us to plot performance of an algorithm under various fading scenarios, or performance of various algorithms under a fading scenario.

1.4 Thesis Outline

In chapter 2, details of Cognitive Radio and Spectrum Sensing has been provided, along with key related research, analysis of the Spectrum Sensing problem and Research gaps have been mentioned. In Chapter 3, details of how the training and testing data is generated has been provided. Brief information about the various Fading Scenarios and Sensing Technique has been mentioned. In Chapter 4, we discuss all the various experiments, their paramaters, and the results along with tables and figures, and the final conclusion. In Chapter 5, we mention the contributions of this thesis along with the limitations and the future scope in this field of study.

Chapter 2

Literature review

2.1 Preamble

This chapter provides background to the Spectrum Sensing problem and the research done by various authors along with the gaps in research in this field of study. This chapter also analyzes the problem and formulates it.

2.2 Background

The requirement for higher bandwidth keeps increasing due to innovation in wireless technologies, and it is apparent that we need an intelligent and dynamic system to resolve this issue. Various campaigns have found that static spectrum access leads to overcrowding in some parts of the spectrum and underutilization in others [16]. This imbalance reduces the effective utilization of the spectrum. We require Cognitive Radio, which enables wireless devices to transmit in spectrum holes as long as the PUs are not transmitting. Cognitive Radios main objective is to use the best spectrum that the PU is not using. Cognitive Radio networks sense opportunities, characterize the environment, determine the best strategy for decisions, and adapt by changing operation parameters. The environment is dynamic, and Cognitive Radio should keep track of the changes in the environment. We mainly focus on the Spectrum Sensing part, the most crucial part of Cognitive Radio Networks. Different techniques for Spectrum Sensing are:

- Energy Detection

- Cyclostationary detection
- Matched Filter Detection

Energy Detection has been widely used as it is the most straightforward technique, it needs less sensing time, and we need no prior knowledge about the PU or its signal. The downside of Energy Detection is the poor performance we obtain when the Signal to Noise Ratio (SNR Value) is low. Cyclostationary detection exploits the second-order periodicity of the modulated signal and provides good results even at low SNR values but is complex to compute. The matched Filter technique maximizes the SNR value of the detected signal, but prior knowledge of the signal is required.

2.3 Key related research

In [16], it is observed that the average occupancy of the spectrum is very low, giving potential to the idea of cognitive Radio and Spectrum Sensing, to maximise the usage of the spectrum band. In [1,5,15,21] compare Machine Learning algorithms to classical algorithms (AND, OR, MRC). Authors have implemented K Means Clustering in [1], and Naïve Bayes, SVM, MLP in [5,21]. Authors in [15] implement various Machine Learning Algorithms, and provide various metrics like Accuracy, Precision, Recall. SVM being a robust algorithm, is used in various papers [5,8,11,12,14,15,17,19,21,23,24,26,29] using various kernels like linear, gaussian, polynomial. Naïve Bayes algorithm, has also been implemented in [5,8,15,21,27].

Deep Learning algorithms have been very popular to solve various issues related to Cognitive Radio. In [28], authors have used a type of Deep Learning model called a Convolutional Neural Network (CNN), mainly used for image recognition and classification. They are good at recognizing complex patterns in the given input (like gradients, shapes, and lines). They have focused on various modulation types (64QAM, B-FM, BPSK, CPFSK, DSB-AM, GFSK, PAM4, QPSK, SSB-AM), and the CNN models try to classify the input data and link it to one of these modulation types. They have also compared the performance of models in different published papers related to modulation types. Authors in [25] mainly emphasize on Context Awareness in Wireless Com-

munications that will improve the efficiency of existing services for which the authors have used Machine Learning and Deep Learning algorithms in various categories like Unsupervised Learning, Supervised Learning, and Reinforcement Learning. The authors have not focused on spectrum sensing. In [3], Deep Reinforcement Learning technique is used, which does not require a labeled dataset and can adapt to the environment with little human interaction and can work dynamically in the environment. It works by rewarding or punishing the model for making decisions. Authors in [19] like [3], use Deep Reinforcement Learning technique to decide whether PU uses the spectrum band. Multiple PUs and a wideband channel are considered. K out of N, SVM, and DL models also have been compared for performance. In [17], Artificial Neural Network, SVM, Decision Tree, and KNN were used to detect PUs presence. The dataset was generated using an Arduino Uno Card and a wireless transmitter, which is a practical dataset. Authors in [22] used: Neural Networks, Expectation-Maximization, and K-Means Clustering. The authors tested models on simulated and real signals to show theoretical and practical performance. Authors in [24] consider SVM, CNN, and Deep Reinforcement Learning models for Cognitive Radio and discuss how several aspects need improvement when applying algorithms for Cooperative Spectrum Sensing.

A few papers have gone in depth to solve a particular issue or considered various other parameters. In [18], authors did not use any Machine Learning algorithms, they have concentrated on SNR Walls, a threshold below which, no matter how long a detector senses, will fail to be robust because it gets tough to distinguish between the h0 and h1 hypotheses. They also discuss what happens on the other side of the SNR Wall, its impact, Spectrum Holes and SNR Walls in space, and how metrics reveal trade-offs and the importance of diversity. In [29] The paper considers that the PU transmits with discrete power levels with set probabilities, and K Means Clustering is used first to label the dataset, and SVM learns from the dataset and the labels to predict if the PU is transmitting or not. In [8] SUs operate under a Hybrid Underlay-Interweave Model, meaning that the SUs can utilize the spectrum when the PU is not transmitting and simultaneously access it along with the PU while abiding by the Interference Temperature. Multiple SVMs, Gaussian Mixture Model (GMM), and Naive Bayes algorithms have been used to decide if the PU is active. Authors in

[7] focus on showing the effects of Malicious Users (MUs) and mitigating their effects. MUs send false data to the Fusion Centre, which would affect the performance of Cooperative Spectrum Sensing. The authors have proposed a Hybrid Boosted Tree Algorithm based on Differential Evolution and the Boosted Tree Algorithm. This model has been compared to other models like the Genetic Algorithm and KNN. The authors have provided detailed results by running simulations with varying parameters like SNR values and population sizes.

Various papers [4,9,25,28] review, condense models and scenarios from other papers. Authors in [9] have published a survey including various types of algorithms under Supervised Learning, Unsupervised Learning, and Reinforcement Learning algorithms from different branches of Machine Learning while authors in [4] focus on Conventional techniques and Advanced techniques, including various methods like Covariance-Based Sensing and Machine Learning techniques.

2.4 Analysis

For Cognitive Radio to work as intended, the algorithms used should come to the correct decision and should come to that decision without taking much time. If the Spectrum Sensing decision is incorrect, the SU may fail to sense a spectrum hole or utilize the spectrum when the PU is still actively transmitting, when an SU should not interfere with the PU. If Spectrum Sensing takes too long, we are wasting time that the SUs could have utilized to occupy the spectrum band.

The downside of Non-cooperative Spectrum Sensing, as the name suggests, is that the decision does not involve cooperation. The environment may lead the SU to consider the wrong decision. Another issue is to find a suitable threshold for the SU. Classical algorithms can detect more efficiently, but now we have to figure out threshold values for multiple SUs at various distances from the PU. An efficient implementation of the MRC algorithm also requires us to have an estimate of the SNR values. Machine Learning algorithms do not have the abovementioned issues.

Machine Learning models have been prevalent to improve the performance of Spectrum Sensing. Various papers have even tried to incorporate Deep Learning models as Deep Learning is more promising [4] [7] [13] [18] [25] [26]. Along with Non-cooperative Spectrum Sensing and Classical

algorithms, this paper aims to compare the performance of a wide variety of Machine Learning algorithms because one algorithm may consistently outperform other algorithms in a few scenarios, and other algorithms may outperform in other scenarios. Performance of Gradient Boosted Algorithms has also been depicted. This paper also compares how these algorithms perform under fading scenarios like Rayleigh Fading, Rician Fading, and Nakagami Fading.

2.5 Problem formulation

The primary goal of Spectrum Sensing is to differentiate between H_0 and H_1 (Fig. 1.1) with high accuracy.

Performance of various algorithms, fading scenarios (h), sensing samples (t), SU numbers, training dataset sizes will be calculated, along with with the most optimal values to give a comperehensive review to show how each of these parameters affect Spectrum Sensing.

2.6 Research gaps

No paper gives a detailed and concise description of performance of Spectrum Sensing considering a wide variety of parameters like Sensing Time, SU numbers, Training Dataset Size and various cooperative Spectrum Sensing algorithms. Performance of Gradient Boosting algorithms like XGBoost, CatBoost, and ADABoost have not yet been depicted.

Chapter 3

Methodology

3.1 Preamble

This chapter provides information on how the training and testing dataset is generated. This chapter also gives description of the various algorithms and fading scenarios that have been used for various experiments.

3.2 Proposed hypothesis/Data Generation/Experiment Description

We consider the PU to be actively transmitting 50% of the time on average ($P(H_1) = 0.5$)). We consider a Cognitive Radio Network with 1 PU and N SUs, distributed evenly at a distance of 500m to 1000m. Each SU senses for time period τ which we can vary and sensing bandwidth $w = 5MHz$. Each SU senses $K = 2\tau w$ samples. Training Dataset can be changed, testing dataset is set at 50000 samples. Noise is a gaussian random variable considered to have zero mean and variance equal to the noise power. Signal is a gaussian random variable considered to have 0 mean and variance equal to the signal power. Signal Channel coefficient can be from Rayleigh Fading, Rician Fading, Nakagami Fading with the desired variance, or be absent, and is multiplied by path loss component.

$$h = g \times d^{\frac{-a}{2}} \tag{3.1}$$

(Eq. 3.1) [21] where g is the fading component of that SU, d is the distance between the SU and the PU, and $a = 4$ (path loss exponent)

After collecting K samples, the estimated normalised energy [21] of an SU is:

$$y = \sum_{k=1}^{K} z(k)^2 \tag{3.2}$$

3.3 Mechanism/Algorithm

Non-cooperative Spectrum Sensing Technique

Non-cooperative Spectrum Sensing is Spectrum Sensing, where each SU independently decides if the PU is actively using the spectrum. It is an inefficient method as the changing environment can cause the SU to predict incorrectly. We select a threshold energy value, and if the sensed energy value is above this value, we consider that the PU is using the spectrum.

We fix the false alarm probability (P_{fa}) [21] and get the below equation.

$$\lambda = \Gamma_u^{-1}(P_{fa}, K/2) \tag{3.3}$$

(Eq. 3.3) Where K is the number of samples and $\Gamma_u(x, n)$ [21] is the upper incomplete Gamma function.

$$\Gamma_u(x, n) = \frac{1}{\Gamma(x)} \times \int_n^{\infty} t^{x-1} e^{-t}\, dx \tag{3.4}$$

Classical Cooperative Spectrum Sensing Techniques

These algorithms involve more than 1 SU, and hence generally perform better because prediction of multiple SUs are being considered. These algorithms should be considered an add-on to the Non-cooperative Spectrum Sensing technique because the computation is almost identical. Only a function is added that decides the rule that has to be applied to individual SU outputs to give the final decision.

- **AND Rule:**

  The final decision is only the logical AND operation [5] applied to all the SU outputs. When all SUs decide that the PU is transmitting, the final output is that the PU is transmitting.

s_i is the outcome of the i_{th} SU, where the total number of SUs is N. Fig (3.5) $\wedge$ is the logical AND operator.

$$v = \begin{cases} 1 & \text{if } (s_1 \wedge s_2 \wedge ...s_N = 1) \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

- **OR Rule:**

  The final decision is only the logical OR operation [5] applied to all the SU outputs. When at least one SU decides that the PU is transmitting, only then the final output is that the PU is transmitting. Fig (3.6) $\vee$ is the logical OR operator.

$$v = \begin{cases} 1 & \text{if } (s_1 \vee s_2 \vee ...s_N = 1) \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

- **Maximum Ratio Combining (MRC):**

  MRC technique [5] multiplies the normalized average SNR value of the SU by the energy value sensed by the SU, so the SUs with the high SNR values have higher influence over the decision. Fig (3.7) This technique increases complexity, and we should know the SNR and energy level correctly.

$$v = \begin{cases} 1 & \text{if } (\sum_{i=1}^{N} w_i y_i \geq \lambda) \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

  Where

$$w_i = \frac{SNR_i}{\sum_j^N SNR_j} \tag{3.8}$$

Machine Learning

Machine Learning is a field of Computer Science that allows computers the ability to learn without being explicitly programmed. In traditional programming. We feed in data and logic to

get the output. In Machine Learning, we feed in data and output, and the machine learns about the problem and comes up with its logic. Machine Learning has countless applications like spam detectors, web search engines, online ads, computer vision, self-driving cars, robotics, and voice assistants. Machine Learning can be of 3 types, Unsupervised Learning, Supervised Learning, and Reinforcement Learning.

- In Supervised Learning, the algorithm learns how to map the labeled data to the labels. We know exactly how many labels or the range of labels we have, and it has lots of real-world applications. These algorithms are not suitable for complex tasks, and we cannot predict accurately if the test data has some variation compared to the training data.

- Unsupervised Learning does not use a labeled dataset. It has three broad applications: Clustering, Dimensionality Reduction, and Association. Clustering is a technique where we label groups of unlabelled data into labels of our own choice. Dimensionality Reduction is a pre-processing stage that aims to simplify the number of features in data when the number of features is too much, making it easier to visualize datasets while preserving the information of the data as much as possible. Association is a method for finding relationships between variables in a dataset that has its applications in marketing where the algorithm understands patterns of customers and suggests other products or offers.

- In Reinforcement Learning, the algorithm is rewarded for doing a desirable behavior and punished for doing an undesirable behavior, and the algorithm learns through trial and error. The algorithm looks for the maximum overall reward to decide correctly.

Deep Learning is a subset of Machine Learning, which tries to mimic the behavior of the brain that allows it to learn data. Deep Learning eliminates some of the pre-processing required and can work on unstructured data like images, text, and audio. Deep Learning understands the data's information, like faces in photos and phrases in a text.

In this paper, we have used various Supervised Learning algorithms and an Artificial Neural Network, which are as follows:

- **Logistic Regression:**

  The Logistic Regression model (logit) estimates the probability of occurrence of an event, and outputs a value between 0 and 1, and if the value is greater than a threshold, then the outcome is Yes, otherwise No. Logit value for a variable x is:

$$logit(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

- **Support Vector Machine:**

  The Support Vector Machine looks for the hyperplane that separates the two data classes and optimizes this margin. A large hyperplane means a clear distinction between the two classes, and the vectors that support this hyperplane are called the Support Vectors. It is not always possible to use a line or a plane to separate data, and this is dealt with by projecting the data to higher dimensions where a plane can separate classes. The separation boundary between the classes can be linear, polynomial, or sigmoid, depending upon the choice of kernel.

- **K Nearest Neighbours:**

  K Nearest Neighbours (K-NN) uses proximity to classify data into labels and can be used for classification or regression problems. Classification problem, the algorithm labels test data based on a majority vote, where the votes are the classes of the K nearest dataset records, where K is a parameter.

- **Naive Bayes:**

  Naïve Bayes method uses the Bayes Theorem to predict the labels.

- **Random Forest:**

  Decision Trees build the model by breaking down the dataset into small subsets, representing them in the form of a tree data structure. A Decision Tree is a structure where each

internal node represents a test on an attribute. Each leaf node holds a label, and each branch represents an outcome. The core algorithm uses Entropy to calculate Homogeneity and Information Gain to construct a Decision Tree. If the sample is homogenous, then the Entropy is 0, and if the sample is equally divided, it has an entropy value of 1. Information Gain is the decrease of Entropy after the data has been split and added to the tree, which the algorithm should maximize, i.e., the most homogenous branches are found. Random Forest Classification consists of many Decision Trees and is an ensemble learning algorithm. Each tree gives its prediction, and the class with the highest votes is the model's output.

- **CatBoost:**

  CatBoost is an open-source software library that provides a gradient boosting framework for Decision Trees. Catboost is popular because of features like native handling of categorical features, fast and scalable, visualization tools for analysis, supporting computation on both CPU and GPU, and is available for Python, R, Java, and C++. Catboost can be used for ranking, classification, and regression.

- **XGBoost:**

  XGBoost stands for Extreme Gradient boosting, is a scalable distributed gradient-boosted decision tree machine learning library, and works for ranking, classification, and regression, like CatBoost. XGBoost provides parallel tree boosting (GBDT). Weights play a crucial role and are assigned to all independent variables, which are then fed into the tree which predicts outcomes. Weights that are mispredicted are increased and fed to the second tree. Trees then ensemble to give an accurate and robust model.

- **ADABoost:**

  ADABoost, short for Adaptive Boosting, is a gradient boosting algorithm. A weak classifier trains on the training data based on weighed samples, where the weight represents how important the sample is to be correctly classified. Initially, all weights are equal. We create

a weak classifier for each variable, and more weight is assigned to the incorrectly classified samples so that their importance increases and they are classified correctly. This process continues until each sample has been classified correctly or the algorithm has reached the maximum iteration level.

- **Multilayer Perceptron:**

  Multilayer Perceptron is a feed-forward neural network because the data flows in the forward direction and consists of only three layers, input layer, hidden layer, and output layer. The input layer receives the data, and the output layer gives the decision. The neurons in the layers learn and train with backpropagation. Backpropagation aims to minimize the cost function and increase accuracy by adjusting the weights and biases which is dependent on the gradients of the cost function with respect to those parameters. After each forward pass, a backward pass is done to adjust the weights. The gradient of the loss is calculated, and is distributed layer by layer backwards.

  In the forward phase (Fig. 3.10) [21], the output for a neuron j in a layer l with weights w, is

$$o_j{}^l = \sigma\left(\sum_i w_{ij}{}^l o_i{}^{l-1}\right) \tag{3.10}$$

  Where $o_i{}^0 = y_i$ and $\sigma(x)$ is the logit (sigmoid) function.

3.4 Analytical validation

Results from papers [5,21] are validated for various parameters and algorithms affecting Spectrum Sensing performance in Cognitive Radio Networks.

Chapter 4

Experiments and results

4.1 Preamble

This chapter provides information on all the various experiments conducted with their parameters, figures and tables, and the conclusion derived from these experiments.

4.2 Experiment design

Here various experiments are conducted showing performance of Spectrum Sensing, they are as follows:

- Performance of All Algorithms
- Performance of Spectrum Sensing under various Fading Scenarios

Table 4.1: Default Parameters

Parameters	**Value**
PU Active Probability	$P(H_1) = 0.5$
Bandwidth	$w = 5MHz$
Sampling Frequency	$\tau = 10MHz$
Number of Samples	$K = 2\tau w = 50$
Number of SUs	$S = 3$
SU distance from PU	[500,750,1000]
Training Dataset Size	250
Testing Dataset Size	50000
Fading Scenario	Rayleigh
Fading Variance	2

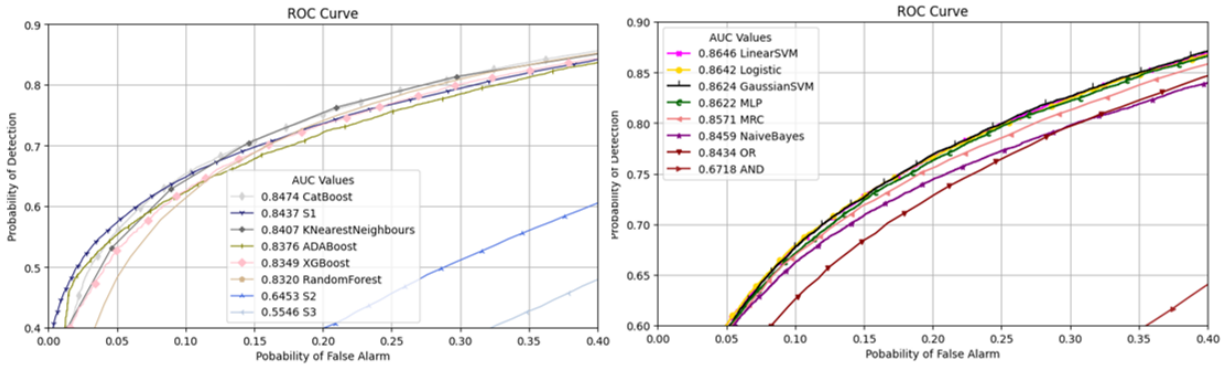

Figure 4.1: ROC Curves for Rayleigh Fading Variance=1

- Performance with respect to varying Sensing Time
- Performance with respect to varying SU numbers
- Performance of MAchine Learning algorithms with respect to varying Training Dataset Size

4.2.1 Experiment 1: Performance of All Algorithms

4.2.1.1 Parameter settings

AUC Values for all algorithms has been compared with fading scenarios for 250 training dataset size, 50000 test dataset, $\tau = 5\mu s$ ($K = 50$ Samples) for $SU = 3$. Various Non-cooperative techniques, Classical Cooperative Techniques, and Machine Learning Techniques have been used for various fading scenarios.

Fig. 4.1-4.9 (a) contains S1, S2, S3 (Non-cooperative Algorithm for each SU), XGBoost, CatBoost, ADABoost, Random Forest Classifier, and K Nearest Neighbours.

Fig. 4.1-4.9 (b) contains Logistic Regression, Linear SVM, Gaussian SVM, MLP, Naive Bayes, and AND, OR, MRC (Classical Cooperative Spectrum Sensing Techniques).

Table 4.2: AUC Values for all algorithms

Algorithms	**Rayleigh**	**Rayleigh**	**Rician**	**Rician**
-	Var=1	Var=2	Var=1	Var=2
Linear SVM	0.8646	0.9376	0.9041	0.9677
Logistic	0.8642	0.9373	0.9043	0.9661
MLP	0.8622	0.9376	0.9059	0.9632
Gaussian SVM	0.8624	0.9353	0.9038	0.9680
OR	0.8434	0.9258	0.8787	0.9579
MRC	0.8571	0.9252	0.9020	0.9631
Naive Bayes	0.8459	0.9250	0.8909	0.9637
XGBoost	0.8349	0.9142	0.8926	0.9645
CatBoost	0.8474	0.9218	0.8856	0.9622
ADABoost	0.8376	0.9203	0.8816	0.9594
RandomForest	0.8320	0.9195	0.8762	0.9598
KNN	0.8407	0.9109	0.8859	0.9610
S1	0.8437	0.9092	0.8924	0.9552
S2	0.6453	0.7275	0.6548	0.7571
AND	0.6718	0.7357	0.6919	0.7642
S3	0.5546	0.6001	0.5524	0.6044

Table 4.3: AUC Values for all algorithms

Algorithms	**Nakagami**	**Nakagami**	**Nakagami**	**Nakagami**	**AWGN**
-	Var=1 M=0.5	Var=1 M=1	Var=1 M=1.5	Var=1 M=2	-
Linear SVM	0.8114	0.8592	0.8871	0.9042	0.9618
Logistic	0.8094	0.8613	0.8753	0.9039	0.9580
MLP	0.7880	0.8618	0.8843	0.8996	0.9611
Gaussian SVM	0.8110	0.8619	0.8896	0.8937	0.9617
OR	0.8006	0.8436	0.8619	0.8756	0.9295
MRC	0.7977	0.8577	0.8856	0.9016	0.9618
Naive Bayes	0.7970	0.8477	0.8778	0.8927	0.9600
XGBoost	0.7945	0.8361	0.8457	0.8919	0.9395
CatBoost	0.7794	0.8484	0.8668	0.8772	0.9503
ADABoost	0.7878	0.8492	0.8683	0.8801	0.9427
RandomForest	0.7655	0.8413	0.8607	0.8495	0.9377
KNN	0.7741	0.8227	0.8728	0.8727	0.9444
S1	0.7812	0.8448	0.8753	0.8923	0.9580
S2	0.6247	0.6453	0.6474	0 .6502	0.6617
AND	0.6401	0.6704	0.6824	0.6905	0.7108
S3	0.5531	0.5526	0.5474	0.5524	0.5522

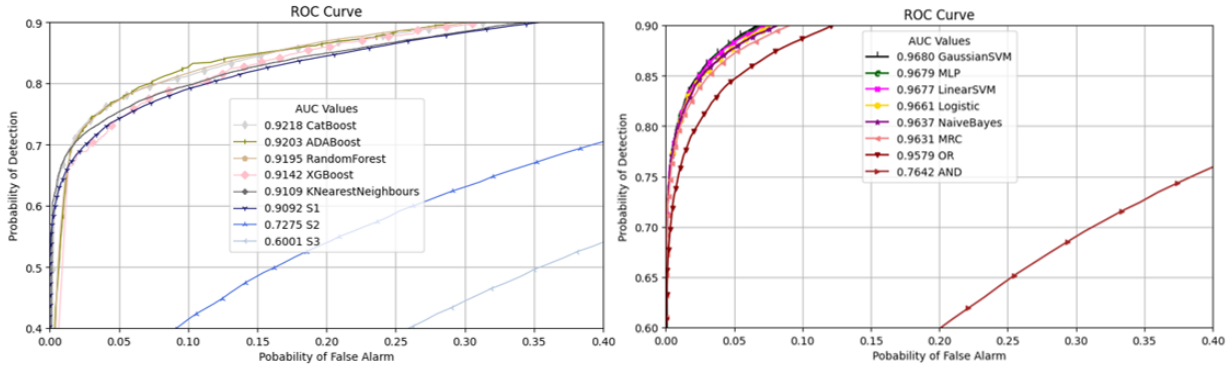

Figure 4.2: ROC Curves for Rayleigh Fading Variance=2

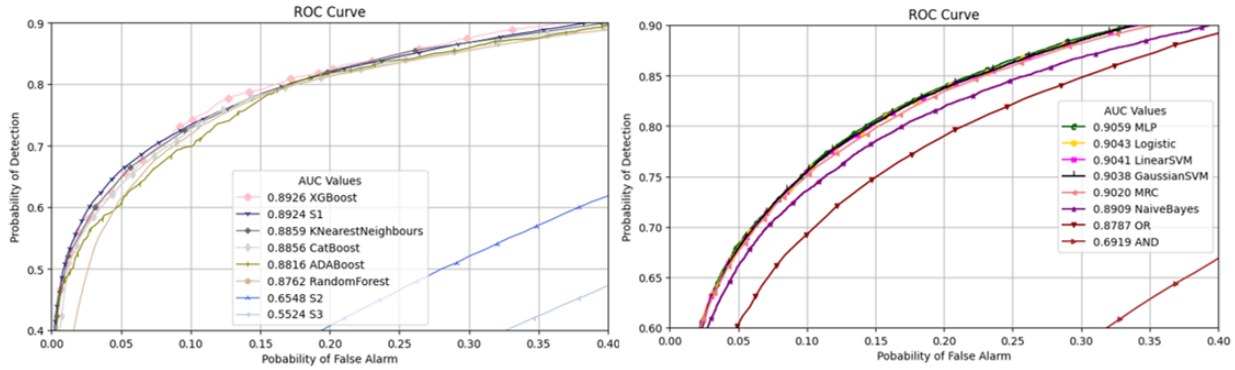

Figure 4.3: ROC Curves for Rician Fading Variance=1

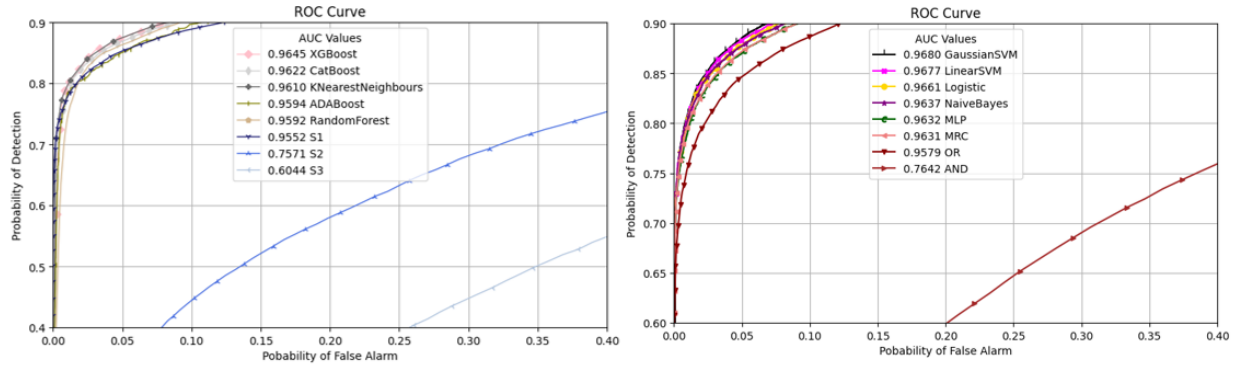

Figure 4.4: ROC Curves for Rician Fading Variance=2

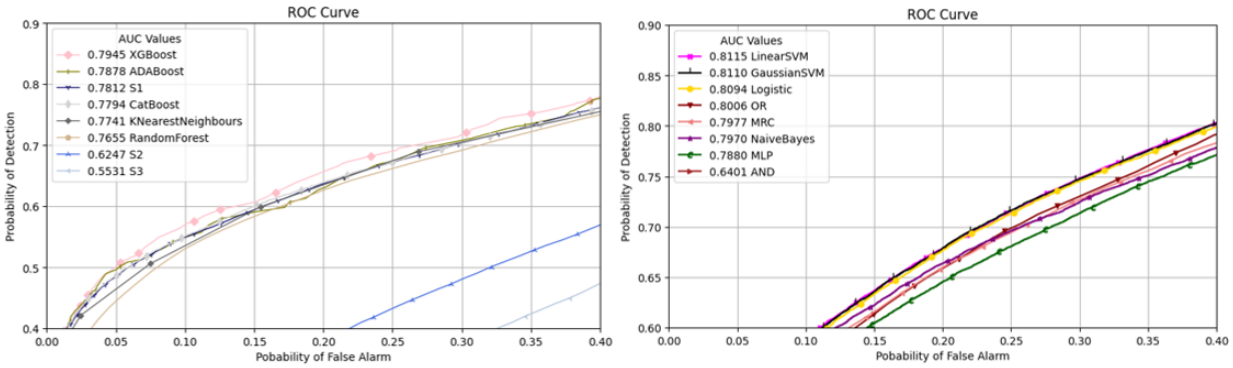

Figure 4.5: ROC Curves for Nakagami Fading Variance=1 M=0.5

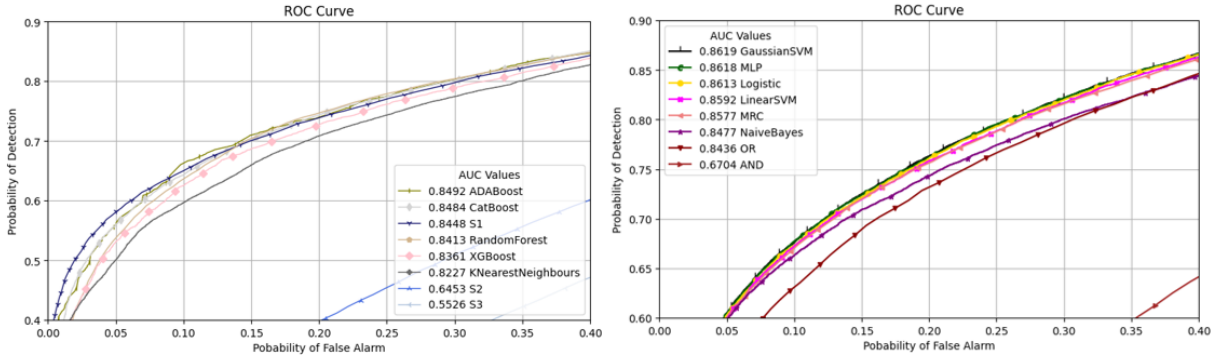

Figure 4.6: ROC Curves for Nakagami Fading Variance=1 M=1

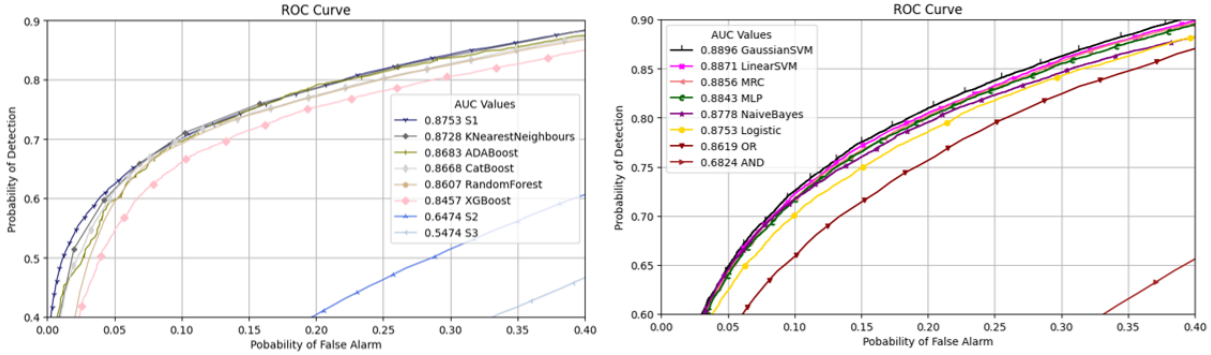

Figure 4.7: ROC Curves for Nakagami Fading Variance=1 M=1.5

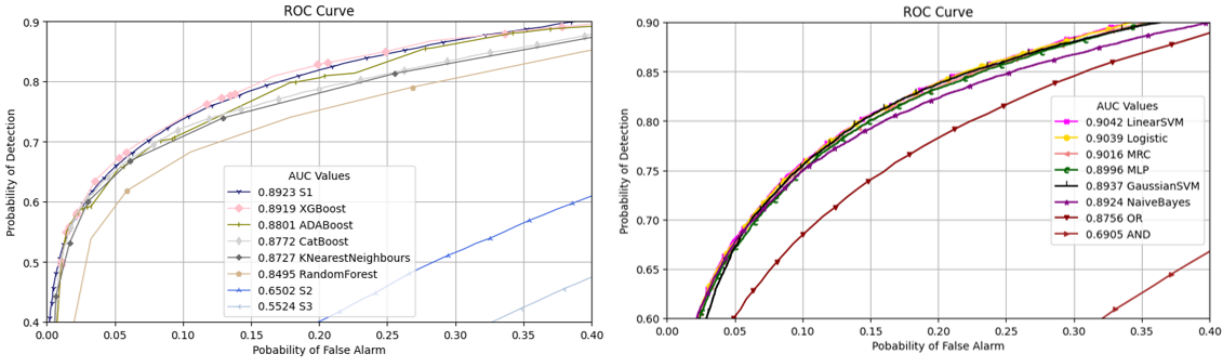

Figure 4.8: ROC Curves for Nakagami Fading Variance=1 M=2

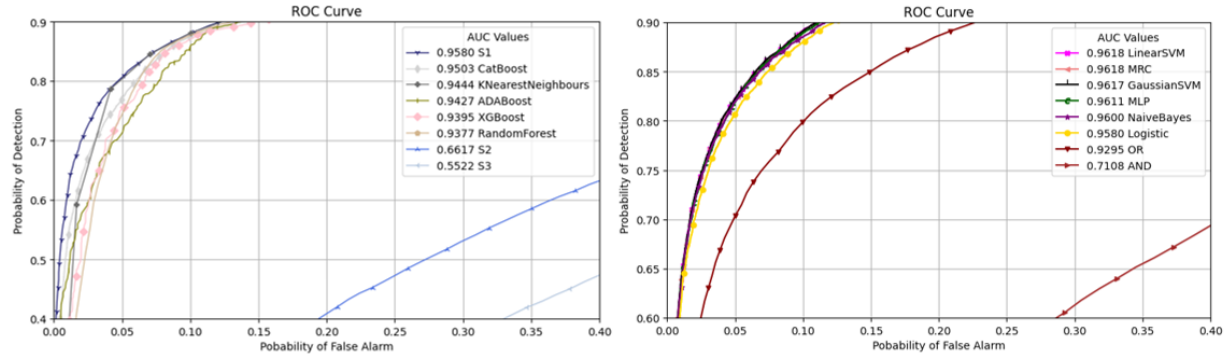

Figure 4.9: ROC Curves for AWGN

4.2.1.2 Results and discussion

With reference to Fig. 4.1-4.9, 4.10 and Table 4.2 and 4.3, talking about Non-cooperative Spectrum Sensing algorithms first, they rank last in performance. Only S1 has acceptable performance, as it is closest to the PU. Perforamance of S3 is only a little better than a random coin flip. The closer an SU is to the PU, the better its performance is. For classical algorithms, AND is the worst performing algorithm of the three. This is because, even if one SU decides that the PU is not transmitting, the final decision is that the PU is not actively using the spectrum which leads to a lot of false alarms. OR gives much better performance because it minimises false alarms, all SUs have to agree that the PU is not transmitting. MRC comes out at the top, with OR method not far behind. This is because MRC is the only algorithm in the test suite that has knowledge of SNR values and it accordingly decides how much priority should be given to each SU in terms of their decision. Coming to Decision Tree based algorithms, they do not have the best performance, but the boosting gradient algorithms consistently outperform Random Forest Classification, proving that they do have potential. K Nearest Neighbours and Naïve Bayes are simpler algorithms and not very robust, hence they hardly outperform powerful algorithms like MLP. In Machine Learning algorithms, the top 4 algorithms are Linear SVM, Gaussian SVM, MLP and Logistic Regression. SVMs are powerful they focus at samples from one label that are closer to the other label (worst case samples). Logistic Regression is simple and robust. MLP does not clearly come at the top, because it is limited by the training dataset size and it only has one hidden layer, yet all four algorithms (Fig. 4.10) are very close in terms of performance, and there is no clear winner, and all four Machine Learning algorithms are suitable, and they consistently outperform Classical Algorithms.

4.2.2 Experiment 2: Performance of All Fading Scenarios

4.2.2.1 Parameter settings

AUC Values for all algorithms has been compared with fading scenarios for 250 training dataset size, 50000 test dataset, $\tau = 5\mu s$ ($K = 50$ Samples) for $SU = 3$ (4.1.1).

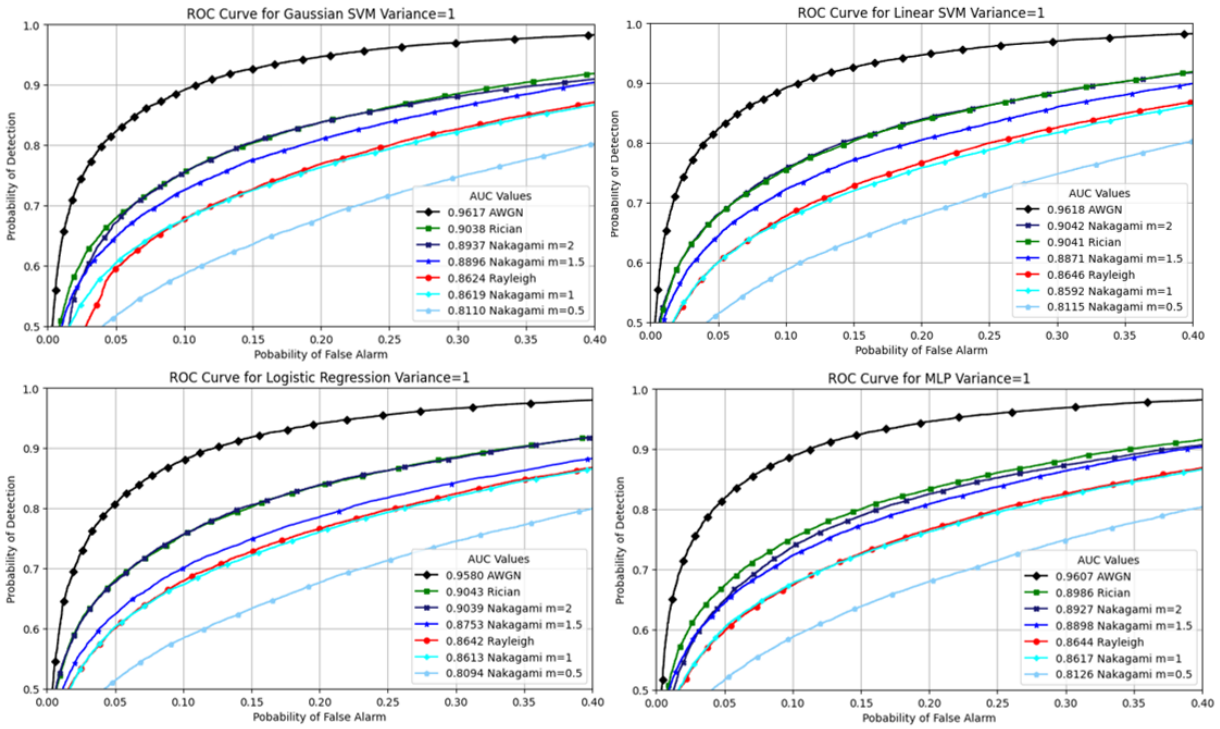

Figure 4.10: ROC Curves depicting performance of ML Algorithms for All Fading Scenarios

AUC Values for Gaussian SVM, Linear SVM, MLP and Logistic Regression have been compared fading scenarios for 250 training dataset size, $\tau = 5\mu s$ ($K = 50$ Samples) for $SU = 3$. Curves for Rayleigh Fading, Rician Fading, Nakagami Fading, and AWGN have been plotted for each algorithm at $Variance = 1$.

4.2.2.2 Results and discussion

With respect to Fig 4.10, Rayleigh Fading (Fig. 4.1 and 4.2) performs the worst, as it assumes that there is no line-of-sight communication, but is the most practical scenario in urban areas. Rician fading (Fig. 4.3 and 4.4) scenario algorithms outperform Rayleigh Fading algorithms because there is line-of-sight communication, so there is a clearer difference in the energy values for H1 and H0. Increasing the M (shape) parameter, for Nakagami Fading (Fig. 4.5-4.8) reduces the effect of fading, which in turn increases the performance. AWGN (Fig. 4.9) outperforms all other fading scenarios, because there is no fading present and this scenario generates the simplest data for algorithms. Fig. 4.10 compares performance of the four best algorithms under all fading scenarios. Rayleigh Fading and Nakagami Fading with parameter M=1 (which is Rayleigh Fading) only have a small difference which can come down to the small variations in the dataset. The difference is larger in MLP only because MLP learns differently and the weighs are different every time, all algorithms have trained on the same dataset corresponding to a fading scenario. Variance=2 (Fig. 4.2, 4.4) is a better scenario compared to Variance=1 (Fig. 4.3, 4.5) because now the random sample has a higher magnitude, and the gain is multiplied with the signal, hence the energy values for H1 are higher, making it easier to differentiate between the two labels.

4.2.3 Experiment 3: Verying Sensing Time

4.2.3.1 Parameter settings

Here τ ranges from $[3\mu s, 7\mu s]$ and hence K ranges from [30,70] samples (Figure 11), using Linear SVM, Gaussian SVM, MLP, Logistic Regression, Naïve Bayes, AND, OR, MRC algorithms. Rayleigh Fading has been used at $Variance = 2$, 250 training dataset size, 50000 test dataset, for

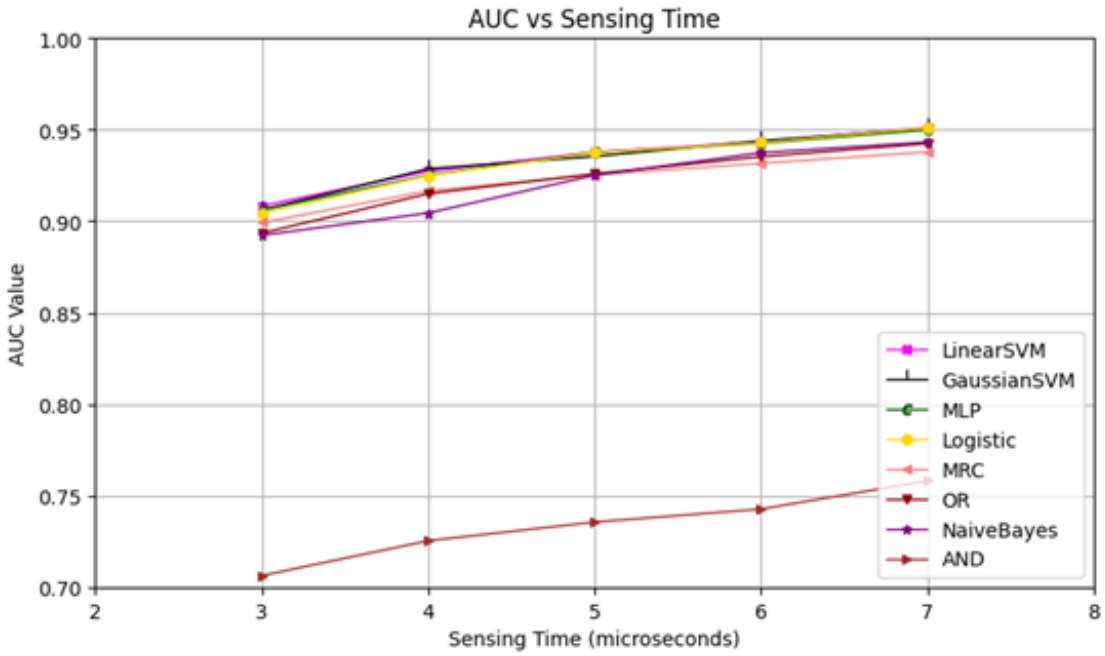

Figure 4.11: AUC vs Sensing Time

Table 4.4: AUC Values with varying number of SUs

Algorithms	**2**	**3**	**4**	**5**	**6**	**6**
Linear SVM	0.9161	0.9376	0.9569	0.9687	0.9766	0.9840
Logistic	0.9144	0.9373	0.9567	0.9687	0.9755	0.9844
MLP	0.9162	0.9376	0.9564	0.9666	0.9730	0.9840
Gaussian SVM	0.9165	0.9353	0.9572	0.9701	0.9792	0.9841
OR	0.9070	0.9258	0.9454	0.9574	0.9669	0.9733
MRC	0.9130	0.9252	0.9446	0.9600	0.9703	0.9785
Naive Bayes	0.9020	0.9250	0.9490	0.9618	0.9749	0.9791
XGBoost	0.8957	0.9142	0.9462	0.9522	0.9689	0.9760
CatBoost	0.9048	0.9218	0.9481	0.9612	0.9726	0.9801
ADABoost	0.8938	0.9203	0.9437	0.9575	0.9691	0.9766
RandomForest	0.8954	0.9195	0.9450	0.9615	0.9710	0.9800
KNN	0.8965	0.9109	0.9430	0.9519	0.9696	0.9683
AND	0.7319	0.7357	0.7387	0.7461	0.7454	0.7486

$SU = 3$.

4.2.3.2 Results and discussion

With respect to Fig 4.11, an average increase of almost 0.01 is seen for every increase in $1\mu s$ (10 Samples), and this value is higher when the number of samples is low, and slowly starts decreasing as we get closer to AUC value 1. The longer we sense, the more information we have to decide, and the clarity between energy values corresponding to the two labels increase. The performance at 70 samples with Rayleigh Fading with $Variance = 2$ is comparable to AWGN performance. We can't keep increasing the Sensing Time because that is time we could have otherwise used to transmit if the spectrum was not being used by the PU. The relationship isn't linear, so we will reach a threshold, which is above 0.95, after which increasing the Sensing Time will not give a substantial increase in performance.

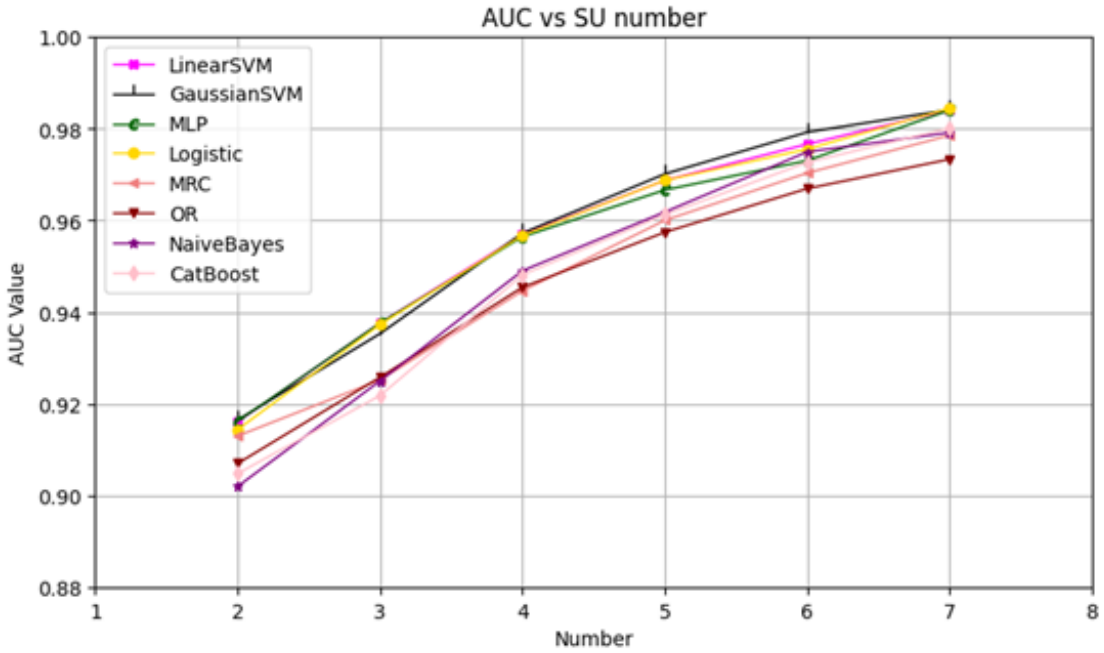

Figure 4.12: AUC vs SU Numbers

Table 4.5: AUC Values for all ML algorithms

Algorithms	**50**	**100**	**250**	**500**	**1000**
Linear SVM	0.9366	0.9344	0.9376	0.9360	0.9367
Logistic	0.9343	0.9344	0.9373	0.9305	0.9373
MLP	0.9201	0.9322	0.9376	0.9365	0.9374
Gaussian SVM	0.9378	0.9264	0.9353	0.9323	0.9368
Naive Bayes	0.9287	0.9152	0.9250	0.9250	0.9261
XGBoost	0.9192	0.9150	0.9142	0.9042	0.9274
CatBoost	0.9240	0.9124	0.9218	0.9213	0.9304
ADABoost	0.8931	0.9024	0.9203	0.9137	0.9280
RandomForest	0.9173	0.9020	0.9195	0.9061	0.9222
KNN	0.9288	0.9097	0.9109	0.9004	0.9228

4.2.4 Experiment 4: Varying numbers of SUs

4.2.4.1 Parameter settings

Experiment implements Linear SVM, Gaussian SVM, MLP, Logistic Regression, Naïve Bayes, CatBoost, OR, MRC for Rayleigh Fading $Variance = 1$. SU range is [2,7] and SUs are distributed evenly between [500m, 1000m]) in the environment from the PU. 250 training dataset size, 50000 test dataset, $\tau = 5\mu s$ ($K = 50$ Samples)

4.2.4.2 Results and discussion

With respect to Fig 4.12 and Table 4.4, The AUC value increases along with increase in SU numbers as there is more sensing data to know whether the PU is active or not. The figure shows that Cooperative Spectrum Sensing provides superior performance, and this increase in performance is highest when the number of SUs is low (for an algorithm, difference in performance of 2 SUs sensing vs 3SUs sensing is substantial), and the difference keeps decreasing as we keep on adding SUs (for an algorithm, difference in performance of 6 SUs sensing vs 7 SUs sensing is negligible).

4.2.5 Experiment 5: Varying Training Dataset Size

4.2.5.1 Parameter settings

AUC Values for all algorithms has been compared with fading scenarios for training dataset size range [50,100,250,500,1000] on the same 50000 testing dataset, $\tau = 5\mu s$ ($K = 50$ Samples) for $SU = 3$.

4.2.5.2 Results and discussion

With reference to Table 4.5, training dataset size varies. Comparing performance of Machine Learning Algorithms. Increasing dataset size does not favour, and performance after training on 50 samples is similar to performance after training for 1000 samples. Only MLP responds to the increasing dataset size, only in the beginning, showing minor improvement. Overall, Increasing or decreasing the dataset does not affect the performance of Machine Learning algorithms, but MLP would not be a less suitable algorithm in a scenario where the machines need to train on less data or where they need to learn quickly.

4.3 Overall conclusion

We have successfully derived results and shown how the performance varies by changing various parameters, we also have found out the best algorithms for Spectrum Sensing. The results depict the order of various fading scenarios based on their performance .The code enables us to create datasets of whatever enviornment using the various parameters and fading scenarios, and we may choose any algorithm for the task of Spectrum Sensing, and we are not limited to the specific parameter values used in the experiments.

With respect to Experiment 1, Gaussian SVM, Linear SVM, MLP and Logistic Regression are the best perfroming algorithms., Graient Boosting Algorithms do not come at the top, but they do outperform Random Forest Classification consistently. For Classical algorithms, MRC is the best performing algorithm, with OR technique not far behind.

With respect to Experiment 2, We obtain best performance in AWGN scenario, as the data is much simpler to learn, followed by Rician Fading and Nakagami Fading depending upon the M value, and then algorithms perform worst in Rayleigh fading scenario. Higher the variance of the Fading, bettwe is the performance.

With respect to Experiment 3, Performance increases with increase in Sensing Time, but the increase starts to flatten as we keep increasing the Sensing Time.

With respect to Experiment 4, Increasing the SU number increases the performance of Spectrum Sensing, the increase in performance starts flattening as we keep on increasing the SU number, and we see a substantial gain in performance when the number of SUs are less.

With respect to Experiment 5, Varying Dataset Size does not give an increase in performance, and most algorithms will perform well even after training on 50 samples.

Chapter 5

Discussions and conclusion

5.1 Contributions

In this report, we have thoroughly discussed effects of various parameters like fading channels, number of SUs, sensing time, training dataset size for Spectrum Sensing by applying a wide variety of algorithms of various types, using metrics like ROC curve and AUC value to compare performance. Results show that Machine Learning algorithms outperform both Classical Cooperative algorithms and Non-cooperative algorithms, with Linear SVM, Gaussian SVM, Logistic Regression and MLP showing the best performance, and Machine Learning algorithms have a lot of potential in this field, as we can get acceptable performance after learning from only 50 samples. The number of SUs dictate the performance, especially when the number of SUs are low. AWGN scenario is most favourable, followed by Rician Fading and Nakagami Fading (order depends on the M parameter value), and then Rayleigh Fading, but we can increase performance by changing other parameters like having more SUs, sensing for longer intervals.

5.2 Limitations

Other Deep Learning Models like Convolutional Neutral Netowrks (CNNs) have not been used. SUs are not randomly distributed, as the performace of various algorithms and fading scenarios have to be compared. Effect of SNR on the Spectrum sensing performance has not been shown.

5.3 Future scope

Interesting aspects that should be considered in the future, is using Deep Reinforcement Learning to have a CRN that learns on itself requiring little human interference. Mobile CRNs are CRNS where either the SUs are mobile or the PU is mobile. Mobile CRNS and the tuning of the parameters to bring the performance close to the performance of static CRNs have a lot of scope as a lot of consumer devices are mobile.

Bibliography

[1] M. A. Abusubiah and S. Khamayseh, Performance of machine learning-based techniques for spectrum sensing in mobile cognitive radio networks, IEEE Access, 10 (2022), pp. 1410–1418.

[2] O. Awe and S. Lambotharan, Cooperative spectrum sensing in cognitive radio networks using multi-class support vector machine algorithms, IEEE Xplore.

[3] M. Aygul, M. Nazzal, and H. Arslan, Deep rl-based spectrum occupancy prediction exploiting time and frequency correlations, IEEE Wireless Communications and Networking Conference (WCNC), (2022), pp. 2399–2494.

[4] S. K. D. Janu, K. Singh, Machine learning for cooperative spectrum sensing and sharing: a survey, John Wiley and Sons, (2021), pp. 1–28.

[5] C. Gattoua, O. Chakkor, and F. Aytouna, An overview of cooperative spectrum sensing based on machine learning techniques, IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), (2020).

[6] A. Ghasemi and E. Sousa, Collaborative spectrum sensing for opportunistic access in fading environments, IEEE Xplore, (2008), pp. 131–136.

[7] N. Gul, S. Kim, S. Ahmed, M. Khan, and J. Kim, Differential evolution based machine learning scheme for secure cooperative spectrum sensing system, Electronics, 14 (2021), pp. 1081–1089.

[8] N. Khalek and W. Hamoud, Learning-based cooperative spectrum sensing in hybrid underlay-interweave secondary networks, Globecom 2020 - 2020 IEEE Global Communications Conference.

[9] S. Khamayseh and A. Halawani, Cooperative spectrum sensing in cognitive radio networks: A survey on machine learning-based methods, Journal of Telecommunications and Information Technology, (2020), pp. 36–46.

[10] V. Krishnakumar, P. Savarinathan, T. Karuppasamy, and A. Jayapalan, Machine learning based spectrum sensing and distribution in a cognitive radio network, International Conference on Computer Communication and Informatics (ICCCI), (2022).

[11] W. Kumar, D. Kandpal, and M. Jain, K-mean clustering based cooperative spectrum sensing in generalized k-mu fading channels, IEEE Xplore, (2016).

[12] Z. Li, W. Wu, X. Liu, and P. Qi, Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks, IEEE Commun., 12 (2018), pp. 2485–2492.

[13] Y. C. Liang, Y. Zeng, E. Peh, and A. Hoang, Sensing-throughput tradeoff for cognitive radio networks, 1326 IEEE Transactions On Wireless Communications, 7 (2008), pp. 1326–1337.

[14] Y. Lu, P. Zhu, D. Wang, and M. Fattouche, Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks, IEEE Wireless Conference and Networking Conference (WCNC 2016) Track 1: PHY and Fundamentals.

[15] A. Mikaeil, B. Guo, and Z. Wang, Machine learning to data fusion approach for cooperative spectrum sensing, , 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 429–434.

[16] K. Patil, R. Prasad, and K. Skouby, A survey of worldwide spectrum occupancy measurement campaigns for cognitive radio, Proceeding of the International Conference on Devices and Communications (ICDeCom), (2011), pp. 1–5.

[17] M. Sabera, A. Rharrasa, R. Saadanea, A. Chehri, N. Hakem, and H. Kharraz, Spectrum sensing for smart embedded devices in cognitive networks using machine learning algorithms, Elsevier, 176 (2020), pp. 2404–2413.

[18] A. Sahai, S. Mishra, and R. Tandra, Spectrum sensing: Fundamental limits.

[19] R. Sarikhani and F. Keynia, Cooperative spectrum sensing meets machine learning: Deep reinforcement learning approach, IEEE Communications Letters, 23 (2020), pp. 1459–1462.

[20] H. Shah and I. Koo, Reliable machine learning based spectrum sensing in cognitive radio networks, Hindawi, (2018), pp. 1–17.

[21] C. Tavares, J. Marinello, M. P. Jr., and T. Abrao, Machine learning-base models for spectrum sensing in cooperative radio networks, IET Commun., 14 (2020), pp. 3102–3109.

[22] Y. Tenorio, A. Guerrero, R. Gonzalez, and S. Boque, Machine learning techniques applied to multiband spectrum sensing in cognitive radios, Sensors, (2019), pp. 1–22.

[23] K. Thilina, K. Choi, N. Saquib, and E. Hossain, Machine learning techniques for cooperative spectrum sensing in cognitive radio networks, IEEE Journal On Selected Areas in Communications IEEE, 31 (2013), pp. 2209–2221.

[24] J. Wang and B. Liu, A brief review of machine learning algorithms for cooperative spectrum sensing, Journal of Physics: Conference Series, (2021), pp. 1–5.

[25] M. Wasilewska, A. Kliks, H. Bogucka, K. Cichon, J. Ruseckas, G. Molis, A. Mackute-Varoneckiene, and T. Krilavacius, Artificial intelligence for radio communication context-awareness, IEEE Access, 9 (2021), pp. 144820–144856.

[26] W. Wu, Z. Li, S. Ma, and J. Shi, Performance improvement for machine learning-based cooperative spectrum sensing by feature vector selection, IET Commun., 14 (2020), pp. 1081–1089.

[27] Y. Xu, P. Cheng, Z. Chen, Y. Li, and B. Vucetic, Mobile collaborative spectrum sensing for heterogeneous networks: A bayesian machine learning approach, IEEE Transactions on Signal Processing, 66 (2018), pp. 5634–5647.

[28] R. Yakkati, R. Yakkati, R. Tripathy, and L. Cenkeramaddi, Radio frequency spectrum sensing by automatic modulation classification in cognitive radio system using multiscale deep cnn, 926 IEEE Sensors Journal, 22 (2022), pp. 926–938.

[29] K. Zhang, J. Li, and F. Gao, Machine learning techniques for spectrum sensing when primary user has multiple transmit powers, IEEE Xplore, (2014), pp. 137–141.