**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. A subset of Machine Learning is Deep 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 the effect of parameters like the distance between PU and SUs, number of SUs, sensing time, different fading scenarios, multiple cooperative spectrum sensing algorithms, classical and Machine Learning techniques, and Non-cooperative Spectrum Sensing algorithms.

Table of Contents

[1.1 Introduction: 4](#_Toc110011402)

[1.2 Background: 5](#_Toc110011403)

[1.3 Objectives: 7](#_Toc110011404)

[1.4 Outline: 7](#_Toc110011405)

[2.1 Related Literature: 1](#_Toc110011406)

[2.2 Fading Channels: 2](#_Toc110011407)

[1. Rayleigh Fading: 3](#_Toc110011408)

[2. Rician Fading: 3](#_Toc110011409)

[3. Nakagami Fading: 3](#_Toc110011410)

[2.3 Non-cooperative Spectrum Sensing: 4](#_Toc110011411)

[2.4 Classical Cooperative Spectrum Sensing Techniques: 4](#_Toc110011412)

[1. AND Rule: 4](#_Toc110011413)

[2. OR Rule: 5](#_Toc110011414)

[3. Maximum Ratio Combining (MRC): 5](#_Toc110011415)

[2.5 Machine Learning: 5](#_Toc110011416)

[1. Logistic Regression: 6](#_Toc110011417)

[2. Support Vector Machine: 6](#_Toc110011418)

[3. K Nearest Neighbours: 6](#_Toc110011419)

[4. Naïve Bayes: 7](#_Toc110011420)

[5. Random Forest: 7](#_Toc110011421)

[6. CatBoost: 7](#_Toc110011422)

[7. XGBoost: 7](#_Toc110011423)

[8. ADABoost: 8](#_Toc110011424)

[9. Multilayer Perceptron: 8](#_Toc110011425)

[2.6 Data Generation: 8](#_Toc110011426)

[3.1 Results: 10](#_Toc110011427)

[1. Performance of Algorithms: 10](#_Toc110011428)

[2. Performance Under Different Fading Scenarios: 12](#_Toc110011429)

[3. Performance with varying Sensing Time: 15](#_Toc110011430)

[4. Performance with varying numbers of SUs: 16](#_Toc110011431)

[5. Performance with training varying dataset size: 17](#_Toc110011432)

[3.2 Final Remarks: 17](#_Toc110011433)

[3.3 References: 18](#_Toc110011434)

# 1.1 Introduction:

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 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, 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 are based upon Non-cooperative Spectrum Sensing, 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 can be classified into three types, Supervised Learning, Unsupervised Learning, and Reinforcement Learning. In Supervised Learning, the dataset the algorithm receives is labeled with output values, 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 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 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. The objective of Cognitive Radio is to obtain the best available spectrum. 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.

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

|  |  |
| --- | --- |
|  | H0 (white space) |
|  | H1 (occupied) |

(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.

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. The downside of Classical CSS algorithms is that they are based on the Non-cooperative Spectrum Sensing approach. 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.

# 1.3 Objectives:

Spectrum Sensing is the important part of Cognitive Radio where the 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

1. Compare performance between:
   * Non-cooperative Spectrum Sensing technique.
   * Classical Cooperative Spectrum Sensing Techniques: AND, OR, and MRC.
   * Machine Learning Cooperative Spectrum Sensing Techniques: 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.
2. To show the effect of Rayleigh Fading, Rician Fading, Nakagami Fading, and AWGN at Variance values 1 and 2 on Spectrum Sensing.
3. Show effect of Sensing Sample Size or Sensing Time, SU numbers, Training Dataset Size for Machine Learning Algorithms on Spectrum Sensing.
4. Find out most optimal parameters and algorithms for Spectrum Sensing based on the above results.

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 Outline:

In Chapter 2, we look at the work related to the field of Cognitive Radio and Spectrum Sensing. We describe the fading channels and the various types of being used in our scenario. We then lay out the process for data generation and all the different parameters that we will tune.

In Chapter 3, we discuss about the various results we have obtained, and present the conclusion.

# 2.1 Related Literature:

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 classification and recognition. 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 Communications 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.

# 2.2 Fading Channels:

Fading is the signal loss in communications which is a random process. It is the variation of the attenuation of a signal because of factors like time, position, objects in the environment, transmission medium, and radio frequency. In an environment, reflectors around the transmitter and receiver exist that create multiple paths that a signal can traverse, and as a result, the receiver sees the superimposition of different versions of the signal that have traveled multiple paths. Each major path behaves as a discrete fading path. Each version will have different factors, attenuation, phase, and delay, that can result in either constructive or destructive interference.  If they were all in phase with each other they would all add together.  Dependent upon the way in which these signals sum together, the signal will vary in strength. If the transmitter or the receiver or objects in the environment move, this can cause the path lengths to change, which will cause relative paths to change, resulting in a change in signal level.

Fading is of two types: Slow Fading and Fast Fading.

1. Slow Fading: It is a result of shadowing by physical barriers, impulse response changing much slower than the signal, low doppler spread, and leads to a drop in SNR value. Various methods like error correcting code and receiver diversity techniques are used to overcome Slow Fading.
2. Fast Fading: It is a result of constructive and destructive interference that may be caused due to multipath fading, high doppler spread, and impulse response changing rapidly with the symbol duration.

There are various fading models some of them being:

## Rayleigh Fading:

Rayleigh Fading assumes that there are one or more major reflected paths from the transmitter to the receiver. In the Rayleigh Fading scenario, there is no single signal path that dominates, and a statistical approach is required to analyze the overall nature of the channel. Rayleigh Fading assumes that the magnitude of the signal travelling in the transmission medium will fade according to the Rayleigh Distribution. Rayleigh Fading is suitable for urban areas where there are various buildings and other object that attenuate and reflect the signal.

Rayleigh Distribution can be generated by using the formula, where U is a random sample from the uniform distribution in the interval (0,1) and is the scale parameter.

## Rician Fading:

Rican Fading assumes that there are one or more major reflected paths from the transmitter to the receiver. Rician Fading occurs when one path, which is typically a line-of-sight signal or a strongly reflected signal, is much stronger than others. Rician Fading assumes that the magnitude of the signal travelling in the transmission medium will fade according to the Rician Distribution. Rayleigh Fading is considered a special case, where there is no line-of-sight signal or a strongly reflected signal.

Rician Distribution can be generated by using the formula.

## Nakagami Fading:

Nakagami Fading occurs when there are relatively large delay-time spreads, with different clusters of reflected waves. Nakagami Fading assumes that the magnitude of the signal travelling in the transmission medium will fade according to the Nakagami Distribution, which is related to the Gamma Distribution. Nakagami Fading has 2 parameters (shape parameter) and (size parameter). For our study, we will keep . Keeping and gives us the Rayleigh distribution.

Nakagami Distribution can be generated by using the formula:

# 2.3 Non-cooperative Spectrum Sensing:

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 (Pfa)and get the below equation.

(5)

Where K is the number of samples and is the upper incomplete Gamma function.

# 2.4 Classical Cooperative Spectrum Sensing Techniques:

(6)

These algorithms are based on the Non-cooperative Spectrum Sensing idea but involve more than 1 SU, and hence generally perform better because predicts 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 applied to all the SU outputs. When all SUs decide that the PU is transmitting, the final output is that the PU is transmitting. is the outcome of the ith SU, where the total number of SUs is N. is the logical AND operator.

(7)

## OR Rule:

The final decision is only the logical OR operation 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. is the logical OR operator.

(8)

## Maximum Ratio Combining (MRC):

MRC technique 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. This technique increases complexity, and we should know the SNR and energy level correctly.

(9)

Where

(10)

# 2.5 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 classified into three types, Supervised Learning, Unsupervised 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:

(11)

## 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.

## Naïve Bayes:

Naïve Bayes method uses the Bayes Theorem to predict the labels. The following relationship is used, given class variable y, and feature vectors x1 to xn.

(12)

term is constant given the input.

can be estimated by Maximum A Posteriori Estimation.

## 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 dependant 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, the output for a neuron in a layer with weights , is

(13)

Where and is the logit (sigmoid) function.

# 2.6 Data Generation:

Data Generation is similar to [1]. We consider the PU to be actively transmitting 50% of the time on average (). 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 which we can vary and sensing bandwidth . Each SU senses 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.

(14)

where is the fading component of that SU, is the distance between the SU and the PU, and (path loss exponent)

After collecting samples, the estimated normalised energy of an SU is:

(15)

# 3.1 Results:

Table 1: AUC Values for all algorithms with fading scenarios for 250 training dataset size,

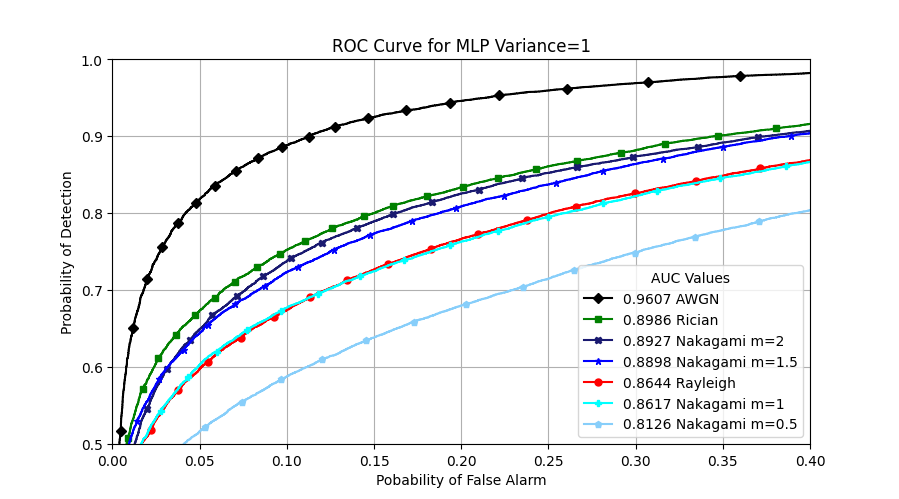
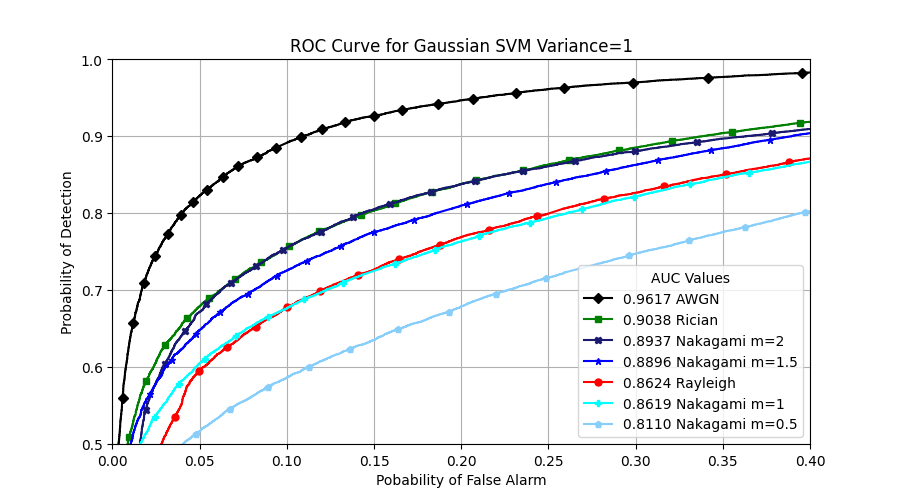
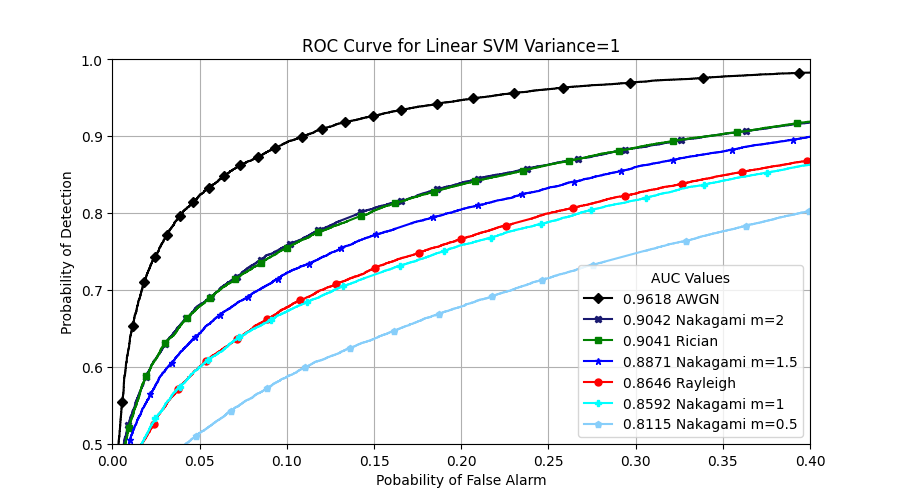
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithms** | **Rayleigh Var=1** | **Rayleigh Var=2** | **Rician Var=1** | **Rician Var=2** | **Nakagami Var=1**  **M=0.5** | **Nakagami**  **Var=1**  **M=1** | **Nakagami**  **Var=1 M=1.5** | **Nakagami**  **Var=1**  **M=2** | **AWGN:** |
| **Linear SVM** | 0.8646 | 0.9376 | 0.9041 | 0.9677 | 0.8114 | 0.8592 | 0.8871 | 0.9042 | 0.9618 |
| **Logistic** | 0.8642 | 0.9373 | 0.9043 | 0.9661 | 0.8094 | 0.8613 | 0.8753 | 0.9039 | 0.9580 |
| **MLP** | 0.8622 | 0.9376 | 0.9059 | 0.9632 | 0.7880 | 0.8618 | 0.8843 | 0.8996 | 0.9611 |
| **Gaussian SVM** | 0.8624 | 0.9353 | 0.9038 | 0.9680 | 0.8110 | 0.8619 | 0.8896 | 0.8937 | 0.9617 |
| **OR** | 0.8434 | 0.9258 | 0.8787 | 0.9579 | 0.8006 | 0.8436 | 0.8619 | 0.8756 | 0.9295 |
| **MRC** | 0.8571 | 0.9252 | 0.9020 | 0.9631 | 0.7977 | 0.8577 | 0.8856 | 0.9016 | 0.9618 |
| **Naïve Bayes** | 0.8459 | 0.9250 | 0.8909 | 0.9637 | 0.7970 | 0.8477 | 0.8778 | 0.8927 | 0.9600 |
| **XGBoosts** | 0.8349 | 0.9142 | 0.8926 | 0.9645 | 0.7945 | 0.8361 | 0.8457 | 0.8919 | 0.9395 |
| **CatBoost** | 0.8474 | 0.9218 | 0.8856 | 0.9622 | 0.7794 | 0.8484 | 0.8668 | 0.8772 | 0.9503 |
| **ADABoost** | 0.8376 | 0.9203 | 0.8816 | 0.9594 | 0.7878 | 0.8492 | 0.8683 | 0.8801 | 0.9427 |
| **RandomForest** | 0.8320 | 0.9195 | 0.8762 | 0.9598 | 0.7655 | 0.8413 | 0.8607 | 0.8495 | 0.9377 |
| **KNN** | 0.8407 | 0.9109 | 0.8859 | 0.9610 | 0.7741 | 0.8227 | 0.8728 | 0.8727 | 0.9444 |
| **S1** | 0.8437 | 0.9092 | 0.8924 | 0.9552 | 0.7812 | 0.8448 | 0.8753 | 0.8923 | 0.9580 |
| **S2** | 0.6453 | 0.7275 | 0.6548 | 0.7571 | 0.6247 | 0.6453 | 0.6474 | 0.6502 | 0.6617 |
| **AND** | 0.6718 | 0.7357 | 0.6919 | 0.7642 | 0.6401 | 0.6704 | 0.6824 | 0.6905 | 0.7108 |
| **S3** | 0.5546 | 0.6001 | 0.5524 | 0.6044 | 0.5531 | 0.5526 | 0.5474 | 0.5524 | 0.5522 |

We by default use 3 SUs, and have 250 training dataset size, 50000 test dataset size, , and fading scenario to be Rayleigh Fading at unless specified otherwise, with AUC metric. For fig 2-10, (i) consists of Non-cooperative Spectrum Sensing techniques for 3 SUs, Gradient Boosting Libraries (XGBoost, CatBoost, ADABoost), K Nearest Neighbours, and Random Forest Classification techniques. (ii) consists of SVMs (Linear and Gaussian), Logistic Regression, Naïve Bayes, MLP, and Classical Spectrum Sensing techniques (AND, OR, MRC).

## Performance of Algorithms:

In reference to Fig. 2-10 and Table 1, talking about Non-cooperative Spectrum Sensing algorithms first, they rank last in performance. S1 only has acceptable performance, as it is closest to the PU. S3s performance 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

Fig 1 (i) (ii) (iii) (iv) ROC Curves depicting performance of ML Algorithms for All Fading Scenarios



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 the two SVMs (Linear and Gaussian), 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. 1) 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.

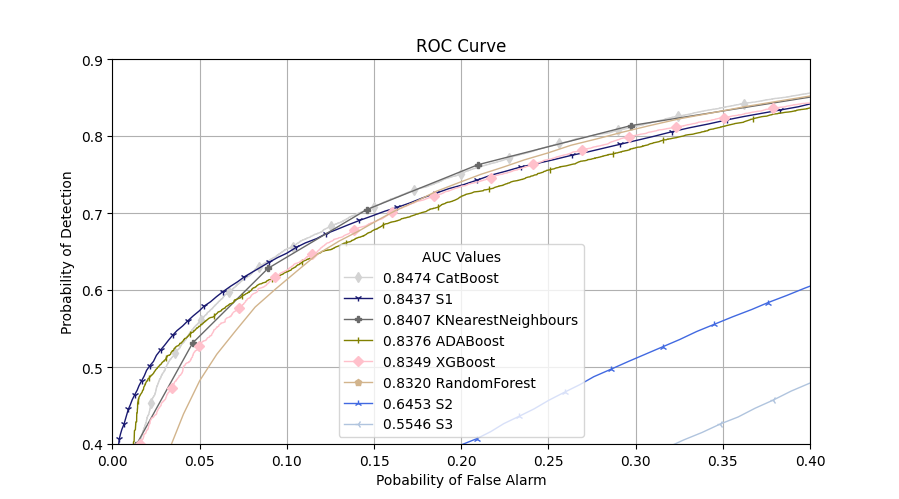
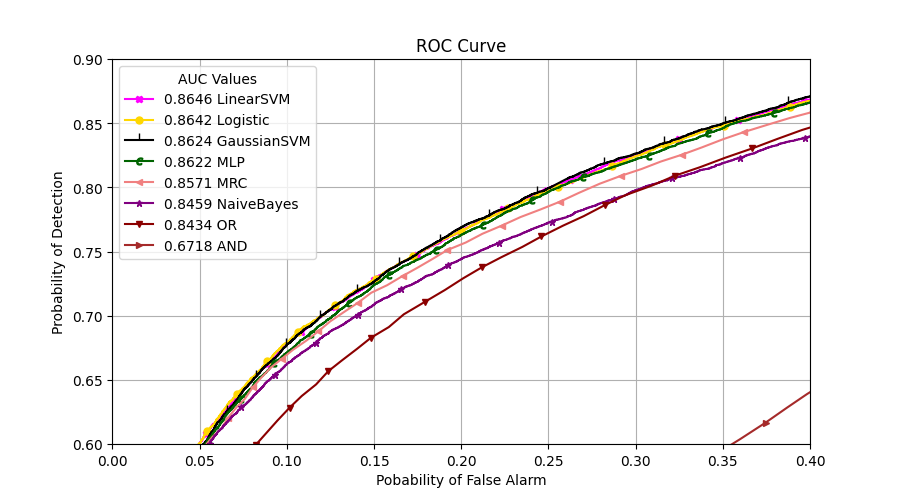


Fig 2 (i) (ii) ROC Curves for Rayleigh Fading

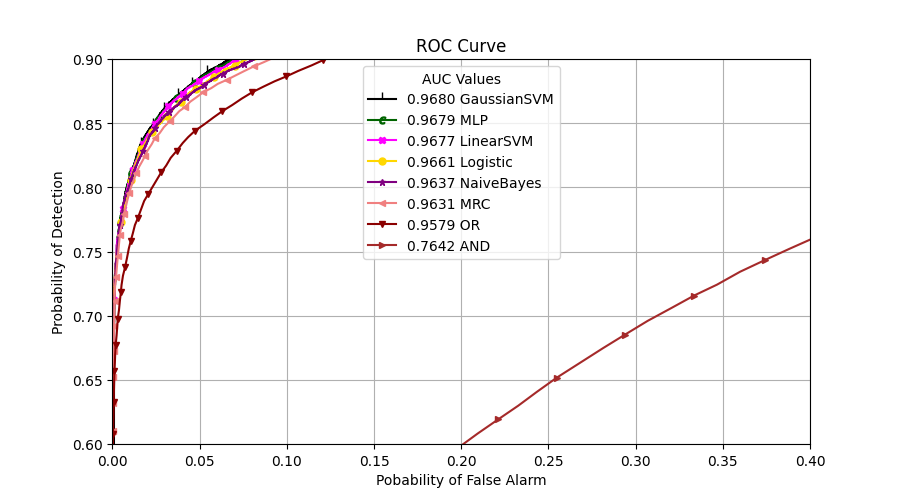


Fig 3 (i) (ii) ROC Curves for Rayleigh Fading

## Performance Under Different Fading Scenarios:

With reference to Table 1, Rayleigh Fading (Fig. 2-3) 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. 4-5) 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 (shape) parameter, for Nakagami Fading (Fig. 6-9) reduces the effect of fading, which in turn increases the performance. AWGN (Fig. 10) outperforms all other fading scenarios, because there is no fading present and this scenario generates the simplest data for algorithms. Fig. 1 compares performance of the four best algorithms under all fading scenarios. Rayleigh Fading and Nakagami Fading with parameter (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. is a better scenario compared to 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.

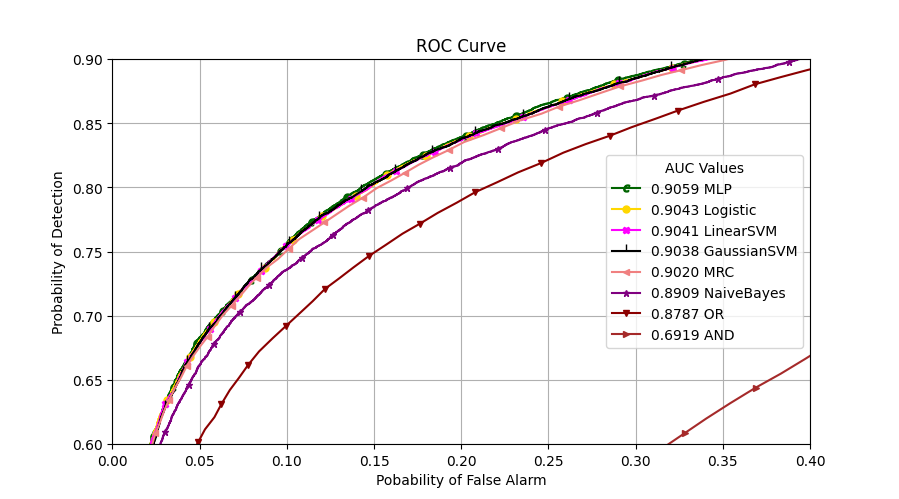
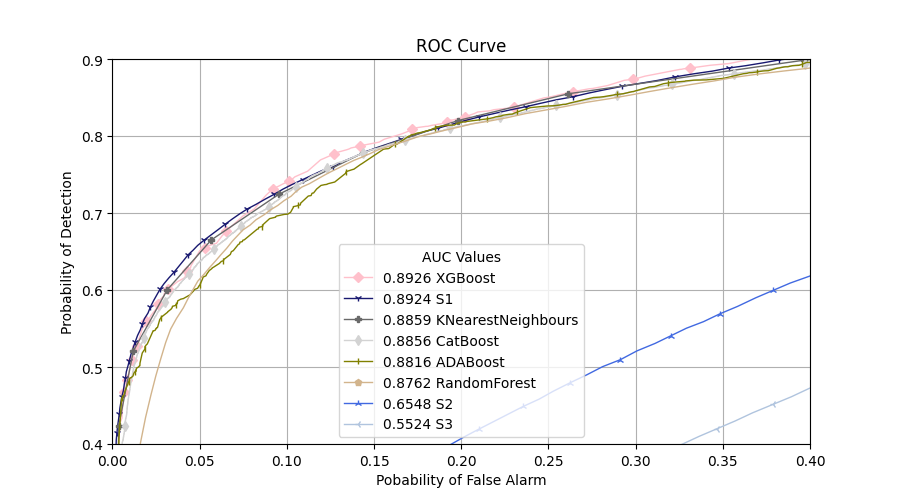


Fig 4 (i) (ii) ROC Curves for Rician Fading

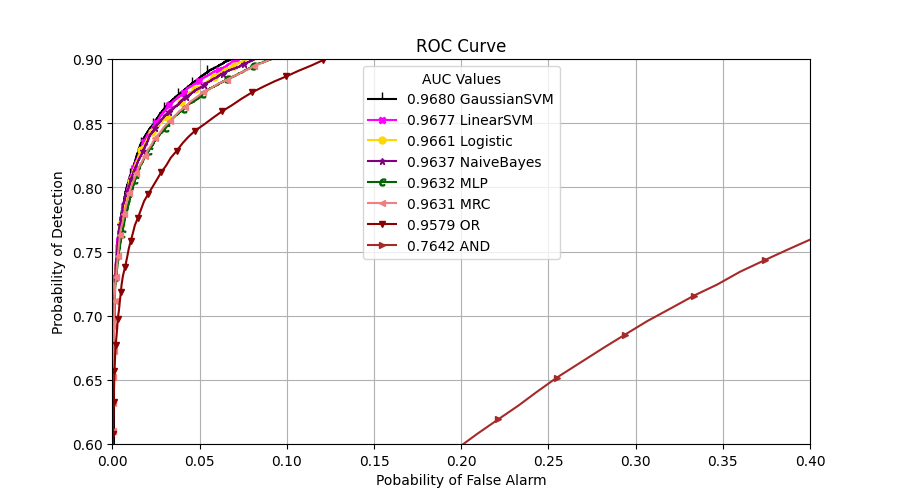
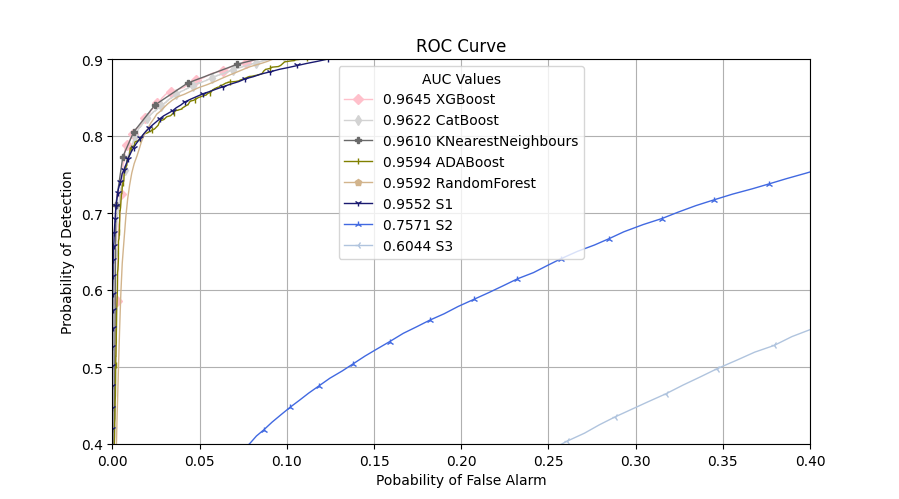


Fig 5 (i) (ii) ROC Curves for Rician Fading

Fig 6 (i) (ii) ROC Curves for Nakagami Fading

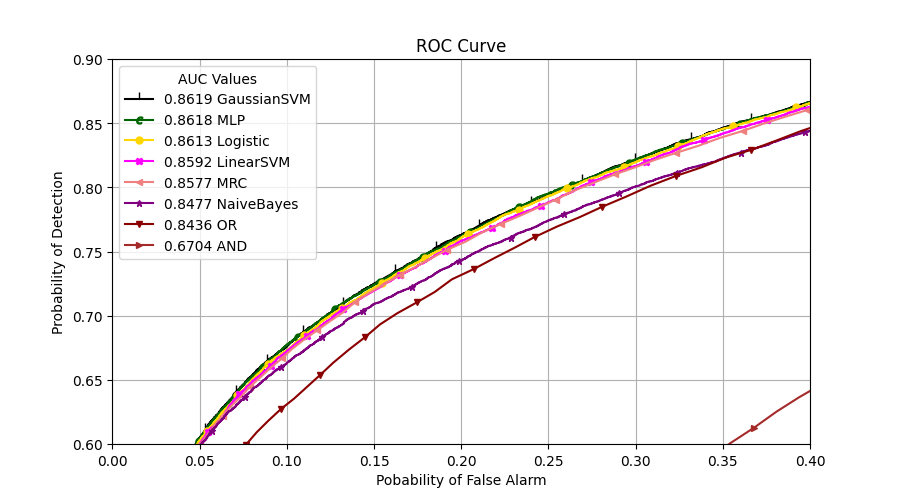
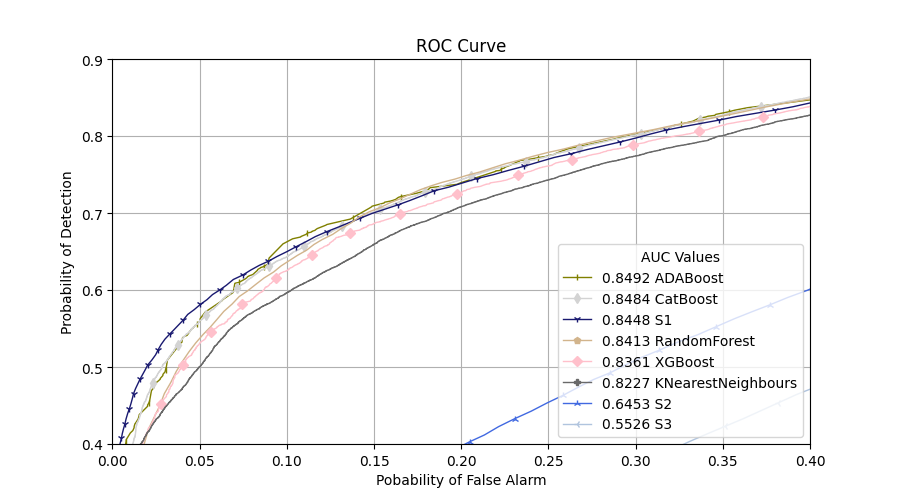
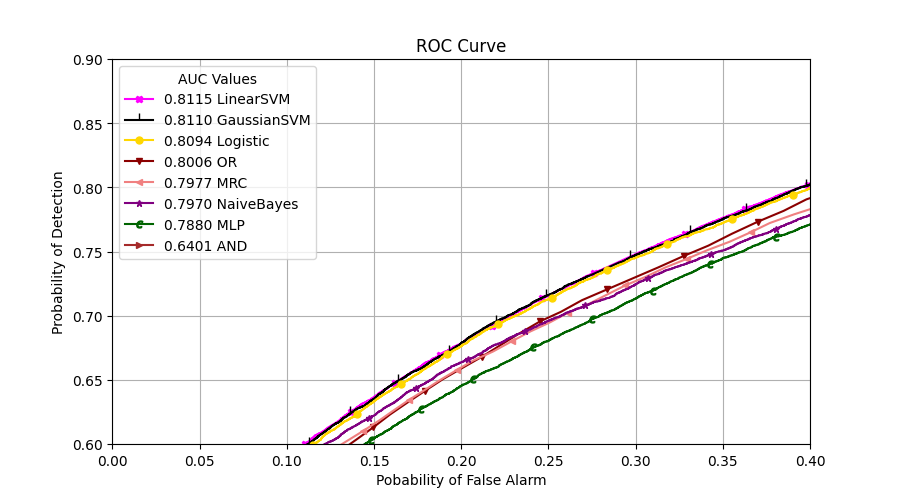
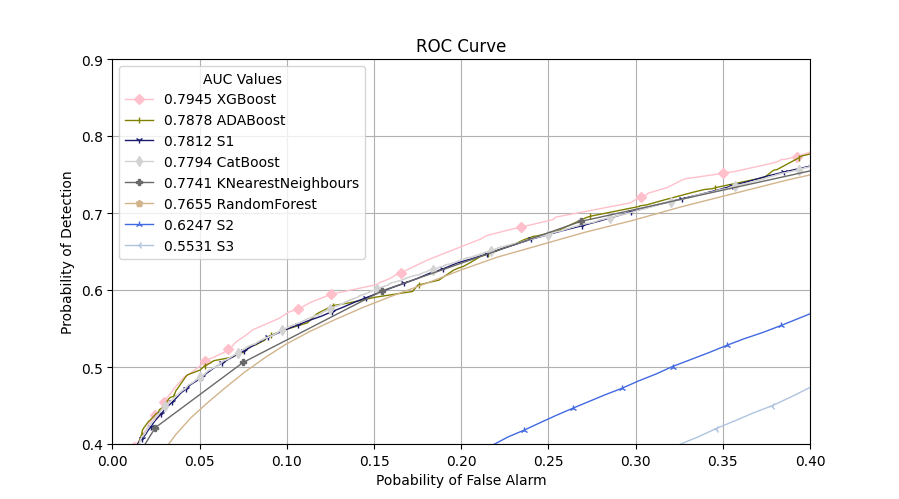


Fig 7 (i) (ii) ROC Curves for Nakagami Fading

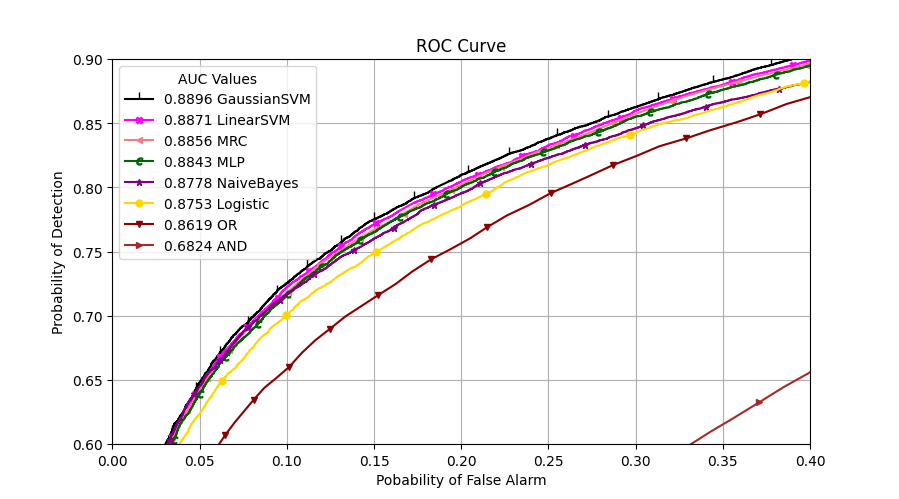
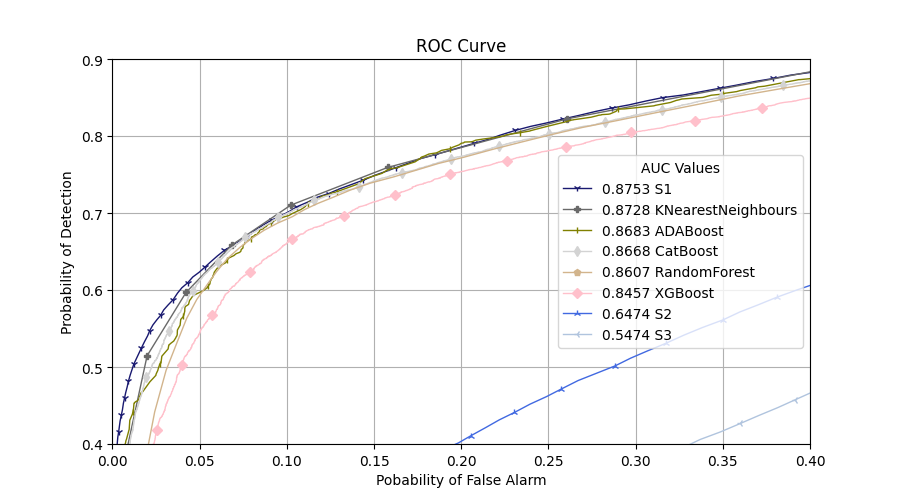


Fig 8 (i) (ii) ROC Curves for Nakagami Fading

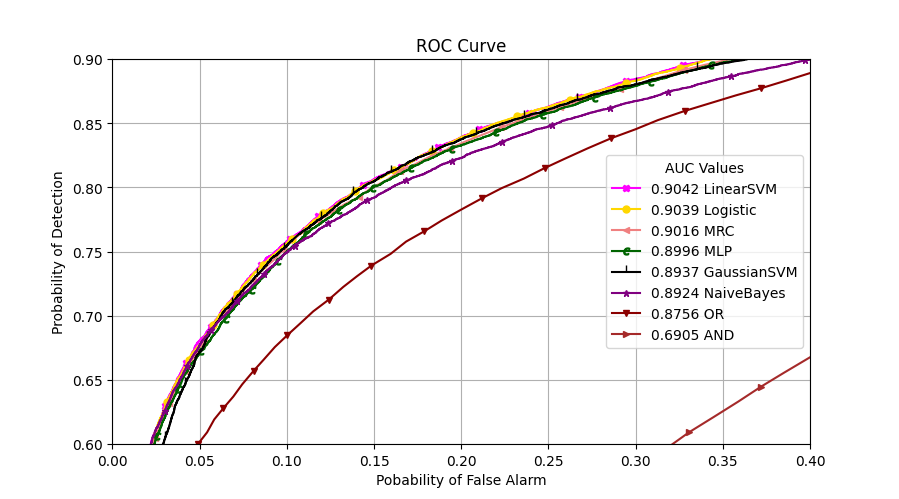
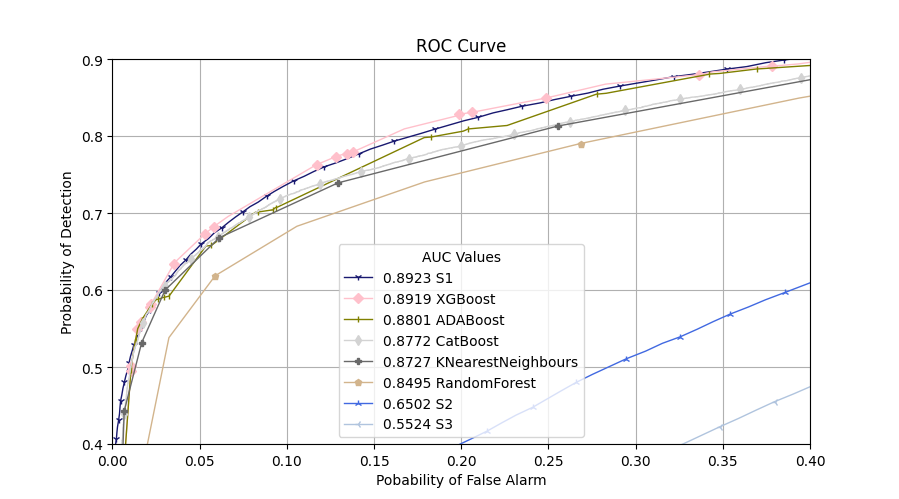


Fig 9 (i) (ii) ROC Curves for Nakagami Fading

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithms** | **2** | **3** | **4** | **5** | **6** | **7** |
| **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 |
| **Naïve 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 |

Table 2: AUC Values with varying number of SUs evenly distributed between [500m, 1000m], for Rayleigh Fading at , 250 training dataset size,

## Performance with varying Sensing Time:

Here ranges from [] and hence ranges from [30,70] samples (Figure 11), using Linear SVM, Gaussian SVM, MLP, Logistic Regression, Naïve Bayes, AND, OR, MRC algorithms. An average increase of almost 0.01 is seen for every increase in 1(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 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.

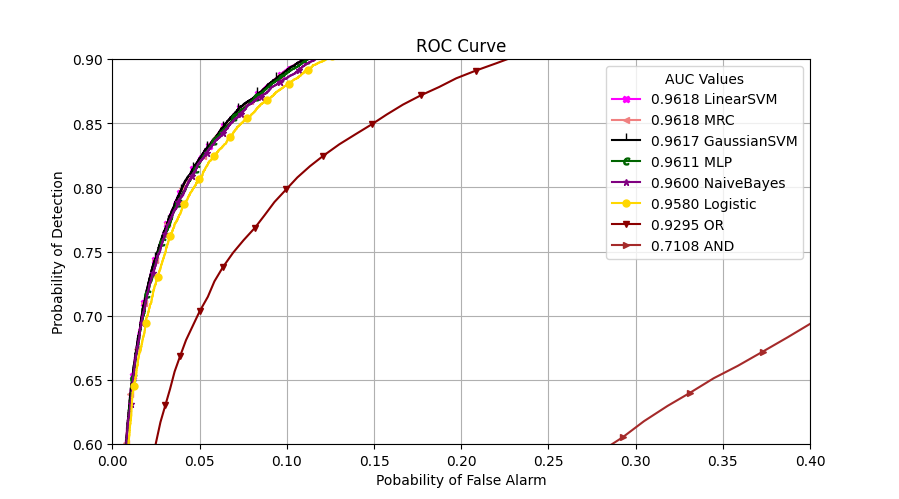
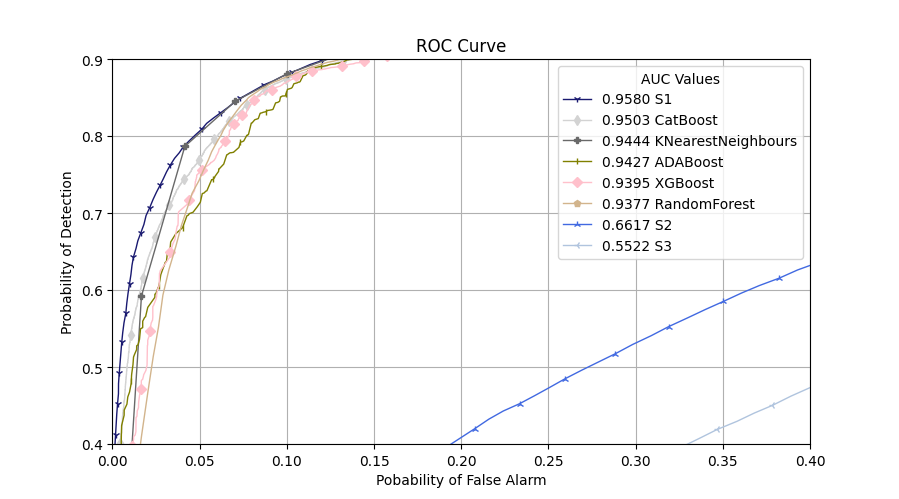
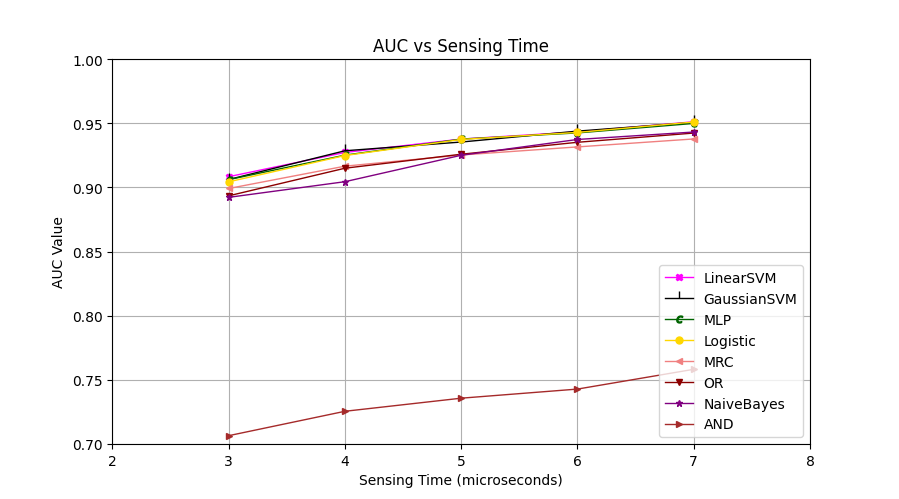


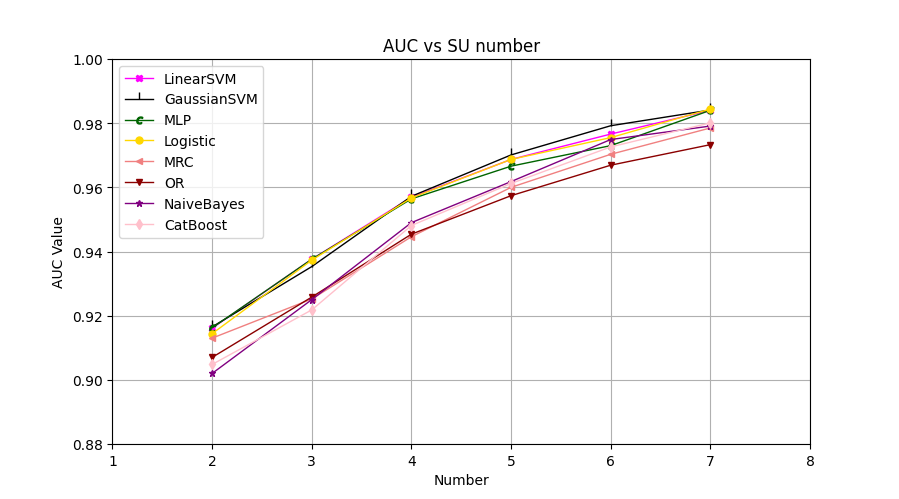
Fig 10 (i) (ii) ROC Curves for AWGN

Fig 11 AUC vs Sensing Time for Rayleigh Fading



## Performance with varying numbers of SUs:

Fig 12 AUC vs SU Numbers for Rayleigh Fading



With Reference to Table 2, and Fig. 12, using Linear SVM, Gaussian SVM, MLP, Logistic Regression, Naïve Bayes, CatBoost, OR, MRC. When there are more SUs ([2,7] SUs, distributed evenly between [500m, 1000m]) in the environment, the AUC value increases 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).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |
| **Naïve 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 |

Table 3: AUC Values for all ML algorithms with varying number of training dataset size for Rayleigh Fading at , ,

## Performance with training varying dataset size:

With reference to Table 3, 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.

# 3.2 Final Remarks:

In this paper, 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 parameter value), and then Rayleigh Fading, but we can increase performance by changing other parameters like having more SUs, sensing for longer intervals.

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 and the to maintain performance in Mobile CRNs where the SUs are mobile while sensing.

# 3.3 References:

1. C. Tavares, J. Marinello, M. Proenca Jr., T Abrao, “Machine learning-base models for spectrum sensing in cooperative radio networks”, IET Commun., 2020, Vol 14 Iss. 18, pp. 3102-3109
2. M. A. Abusubiah, S. Khamayseh, “Performance of Machine Learning-Based Techniques for Spectrum Sensing in Mobile Cognitive Radio Networks”, IEEE Access, Volume 10, 2022, pp. 1410-1418
3. Sahai, S. Mishra,R Tandra, “Spectrum Sensing: Fundamental Limits”
4. R. Yakkati, R. Yakkati, R. Tripathy, L Cenkeramaddi, “Radio Frequency Spectrum Sensing by Automatic Modulation Classification in Cognitive Radio System Using Multiscale Deep CNN”, 926 IEEE Sensors Journal, Vol. 22, No. 1, Jan 2022, pp. 926-938
5. M. Wasilewska, A. Kliks, H. Bogucka, K. Cichon, J. Ruseckas, G. Molis, A. Mackute-Varoneckiene, T. Krilavacius, “Artificial Intelligence for Radio Communication Context-Awareness”, IEEE Access, Volume 9, 2021, pp. 144820-144856
6. V. Krishnakumar, P. Savarinathan, T. Karuppasamy, A. Jayapalan, “Machine Learning based Spectrum Sensing and Distribution in a Cognitive Radio Network”, 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, Coimbatore, India
7. M. Aygul, M. Nazzal, H. Arslan, “Deep RL-Based Spectrum Occupancy Prediction Exploiting Time and Frequency Correlations”, 2022 IEEE Wireless Communications and Networking Conference (WCNC), pp. 2399-2404
8. Y. C. Liang, Y. Zeng, E. Peh, A. Hoang, “Sensing-Throughput Tradeoff for Cognitive Radio Networks”, 1326 IEEE Transactions On Wireless Communications, Vol. 7, No. 4, 2008, pp. 1326-1337
9. Ghasemi, E. Sousa, “Collaborative Spectrum Sensing for Opportunistic Access in Fading Environments”, IEEE Xplore, 2008, pp. 131-136
10. K. Zhang, J. Li, F, Gao, “Machine Learning Techniques for Spectrum Sensing When Primary User Has Multiple Transmit Powers”, IEEE Xplore, 2014, pp. 137-141
11. N. Khalek, W. Hamouda, “Learning-Based Cooperative Spectrum Sensing in Hybrid Underlay-Interweave Secondary Networks”, Globecom 2020 - 2020 IEEE Global Communications Conference
12. C. Gattoua, O. Chakkor, F. Aytouna, “An overview of Cooperative Spectrum Sensing based on Machine Learning Techniques”, 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)
13. M.Sabera, A. Rharrasa, R. Saadanea, A. Chehri, N. Hakem, H. Kharraz, “Spectrum Sensing for Smart Embedded Devices in Cognitive Networks using Machine Learning Algorithms”, Elsevier, Volume 176, 2020, pp. 2404-2413
14. Y. Xu, P. Cheng, Z. Chen, Y. Li, B. Vucetic. “Mobile Collaborative Spectrum Sensing for Heterogeneous Networks: A Bayesian Machine Learning Approach”, IEEE Transactions on Signal Processing, Vol 66, No. 21, 2018, pp. 5634-5647
15. H. Shah, I. Koo, “Reliable Machine Learning Based Spectrum Sensing in Cognitive Radio Networks”, Hindawi, Volume 2018, pp. 1-17
16. Y. Lu, P, Zhu, D. Wang, 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
17. K. Thilina, K. Choi, N. Saquib, E. Hossain, “IEEE Wireless Conference and Networking Conference (WCNC 2016) Track 1: PHY and Fundamentals”, IEEE Journal on Selected Areas in Communication, VOL. 31, NO. 11, NOVEMBER 2013, pp. 2209-2221
18. R. Sarikhani, F. Keynia, “Cooperative Spectrum Sensing Meets Machine Learning: Deep Reinforcement Learning Approach”, IEEE Communications Letters, Vol 23, No. 7. 2020, pp. 1459-1462
19. Mikaeil, B. Guo, 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
20. N. Gul, S. Kim, S. Ahmed, M. Khan, J. Kim, “Differential Evolution Based Machine Learning Scheme for Secure Cooperative Spectrum Sensing System”, Electronics 2021, 10, 1687, pp. 1-25
21. W. Wu, Z. Li, S. Ma, J. Shi, “Performance improvement for machine learning-based cooperative spectrum sensing by feature vector selection”, IET Commun., 2020, Vol. 14 Iss. 7, pp. 1081-1089
22. W. Kumar, D. Kandpal, M. Jain, “K-mean Clustering based Cooperative Spectrum Sensing in Generalized κ-µ Fading Channels”, 2016, IEEE Xplore
23. O. Awe, S. Lambotharan, “Cooperative Spectrum Sensing in Cognitive Radio Networks Using Multi-class Support Vector Machine Algorithms”, IEEE Xplore
24. Z, Li, W. Wu, X. Liu, P. Qi, “Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks”, IET Commun., 2018, Vol. 12 Iss. 19, pp. 2485-249
25. Y. Tenorio, A. Guerrero, R. Gonzalez, S. Boque, “Machine Learning Techniques Applied to Multiband Spectrum Sensing in Cognitive Radios”, Sensors 2019, 19, 4715, pp. 1-22
26. J. Wang, B. Liu, “A Brief Review of Machine Learning Algorithms for Cooperative Spectrum Sensing”, 2021, Journal of Physics: Conference Series, pp. 1-5
27. S. Khamayseh, 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
28. D. Janu, K. Singh, S. Kumar, “Machine learning for cooperative spectrum sensing and sharing: a survey” John Wiley and Sons, 2021, pp. 1-28
29. Patil, K., 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), pp. 1–5, February 2011