SPECTRUM SENSING FOR DATA TRANSMISSION IN WIRELESS COMMUNICATION

A Main Project Report

Submitted to the FACULTY of ENGINEERING of

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In partial fulfillment of the requirements, for

the award of the Degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

K. Meghana
(19481A0493)
K. Leela Venkata Satya Sai Charan
(19481A04B8)

M. Sivaiah
(20485A0412)
K. Mahanth Venkata Sumanth
(19481A04B4)

Under the Guidance of

Sri E. Vargil Vijay

Assistant Professor

Department of Electronics and Communication Engineering

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU – 521356
ANDHRA PRADESH 2022-23

Department of Electronics and Communication Engineering

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU – 521356

CERTIFICATE

This is to certify that the project report entitled "**Spectrum Sensing For Data Transmission In Wireless Communication**" is a bonafide record of work carried out by **K. Meghana (19481A0493), M. Sivaiah (20485A0412), K. Leela Venkata Satya Sai Charan (19481A04B8), K. Mahanth Venkata Sumanth (19481A04B4)** under my guidance and supervision in partial fulfillment of the requirement for the awards of the degree of Bachelor of Technology in **Electronics and Communication Engineering** of **Seshadri Rao Gudlavalleru Engineering College** affiliated to **Jawaharlal Nehru Technological University, Kakinada.**

(Sri E. Vargil Vijay)
Project Guide

(DR. Y. RAMA KRISHNA)
Head of the Department

Acknowledgement

We are very glad to express our deep sense of gratitude to **E. VARGIL VIJAY,** Assistant Professor, Electronics and Communication Engineering for guidance and cooperation in completing this main project. We convey our heartfelt thanks to him for inspiring assistance till the end of our main project.

We convey our sincere and indebted thanks to our beloved Head of the Department **Dr. Y. RAMA KRISHNA**, for his encouragement and help for completing our project successfully.

We also extend our gratitude to our Principal **Dr. G. V. S. N. R. V. PRASAD**, for thesupport and for providing facilities required for the completion of our project.

We impart our heartfelt gratitude to all the Lab Technicians for helping us in all aspects related to our project.

We thank our friends and all others who rendered their help directly and indirectly to complete our project.

K. Meghana (19481A0493)

M. Sivaiah (20485A0412)

K.L.V.S.S Charan (19481A04B8)

K.M.V Sumanth (19481A04B4)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The main focus of this project is Spectrum sensing for efficient data transfer in wireless communications. Currently, there is a need for an effective wireless communication system because the serviceable spectrum is not sufficient due to exponential growth in the number of wireless device users. One approach for effectively using the spectrum is through Radio frequency (spectrum) sensing. Cognitive radio relies on Radio frequency sensing to detect available spectrum for better usage and to reduce harmful interference with licensed users. For spectrum sensing, there are currently established techniques based on energy detection, but the development of machine learning has made spectrum sensing more efficient. A machine learning-based model can perform better than traditional methods. To identify the existence of a licensed user, we suggest a machine learning based spectrum sensing technique employing a Convolutional Neural Network model in a Gaussian environment. Such that if the spectrum is not being utilized by the licensed user then the secondary user can benefit from it. This methodology will be very helpful in resolving the spectrum scarcity problem in wireless communications.

Key Words: *Spectrum sensing, Spectrum utilization, Energy detection, CNN.*

CHAPTER-1

1.1 Background

Presently, due to the increase in the number of users in wireless networks against limited radio spectrum, problems have arisen in spectrum management. The radio spectrum is underutilized in the static approach to spectrum management. The conventional allocation of the spectrum is based on the proper utilization of the spectrum. However, it is inflexible because each wireless operator has to be given a license to operate at a certain frequency. Above all, it is becoming difficult to find vacant bands in the radio spectrum to deploy new services, as most portions of the spectrum have already been allocated. To overcome this problem, it has become necessary to find out technological means for improved utilization of the spectrum to create opportunities for dynamic spectrum access. Currently, the 'Cognitive Radio' (CR) technology is the best technology available to improve the spectrum utilization in wireless communications. The CR technology, which is endowed with spectrum awareness, thrusts itself as a suitable candidate to solve the problem of the scarce radio resources. Such technology can be integrated with the next cellular wireless standards.

Unlicensed frequency bands in India

Unlicensed Frequency Ranges in India	Applications
50-200 kHz	Very low power devices
13553-13567 kHz	Very low power radio frequency devices, Indoor only
26.957 MHz-27.283 MHz	Low power wireless equipment
335 MHz	Low power wireless equipment for the remote control of cranes
402-405 MHz	Medical RF wireless devices with channel emission bandwidth within 300 KHz
865-867 MHz	Low power wireless device with 200 kHz carrier bandwidth.
2400 MHz-2483.505 MHz	Low power wireless equipment with spectrum spread of 10 MHz or higher.
5150 MHz-5350 MHz	Low power equipment for Wireless Access Systems

	indoor only.
5725 MHz-5825 MHz	Low power equipment for Wireless Access Systems indoor only
5825 MHz-5875 MHz	Low power equipment with a spectrum spread of 10 MHz or higher.

Table. 1.1: Unlicensed frequency bands in India

The CR is used to provide reliable communication between the users at any time by effectively utilizing the spectrum. The key function of the CR is the spectrum sensing which opportunistically detects the unutilized spectrum, which is licensed to the primary user (PU). The three types of channel information used by the cognitive radio are overlay, underlay, and interweave.

i) Underlay Access

The SU may transmit simultaneously with the PU over the same channel. However, the transmitted power should not exceed a certain threshold in order to keep the interference on PU below a tolerable value.

ii) Overlay Access

The SU may transmit simultaneously with the PU on the same channel up to its maximum power, but at the cost of playing a role of relay between two or more PUs. In this case, the SU sends its data while relaying the PUs. This kind of access requires high level of cooperation between PUs and SUs, which may expose the PUs privacy.

iii) Interweave Access

SU is allowed to transmit using its maximum power only when PU is absent. In this model, the CR system has to monitor periodically the radio spectrum and detect the activity of possible PUs in different parts of the spectrum; Then, it allows communication over spectrum holes (ie., temporary space-time-frequency voids) and ceases transmission once a PU activity is declared.

Spectrum Sensing

Radio signal detection in a specific frequency range is known as spectrum sensing. In dynamic spectrum access systems, where wireless devices can opportunistically access unused frequency bands, this technique is extremely important for facilitating optimal use of spectrum resources. Spectrum sensing can be used in the context of data transfer to identify channels with sufficient bandwidth. Primary transmitter detection refers to the process of identifying the presence or absence of a primary transmitter signal in a particular frequency band.

Spectrum sensing can be accomplished by a variety of methods, such as energy detection,

cyclostationary feature identification, matched filtering, and compressive sensing. Different methods have different complexity levels, precision levels, and computational needs. To identify primary user, we propose a Convolutional Neural Network (CNN) model operating in a Gaussian environment for spectrum sensing. There are mainly two types of spectrum sensing namely narrowband and wideband sensing.

Narrowband Spectrum Sensing

Narrowband spectrum sensing is used for finding the occupancy status of a single PU licensed band. The term narrowband implies that the bandwidth of the signal is sufficiently small such that the channel frequency response can be considered flat. In other words, the bandwidth of our interest is smaller than the coherence bandwidth that represents the maximum bandwidth over which the channel response is flat. The task of narrowband spectrum sensing is to decide whether the narrow slice of PU spectrum is available for secondary usage or not. Basically, the task of narrowband spectrum sensing is to decide between two hypotheses HO and H1 defined as:

Wideband Spectrum Sensing

Wideband spectrum sensing, multiple bands are sensed for finding spectrum opportunities. In general terms, the wideband consists of a large number of narrow sub-bands. Here, we aim to sense a frequency bandwidth that exceeds the coherence bandwidth of a channel. For example, wideband spectrum sensing aims at finding spectrum opportunities over the whole ultra-high frequency (UHF) TV band (between 300 MHz to 3 GHz). To do this, one should note that narrowband sensing techniques cannot be directly used for the wideband case. This is because these techniques make a single binary decision for the whole spectrum and thus cannot identify individual spectrum holes that lie within the broad frequency spectrum.

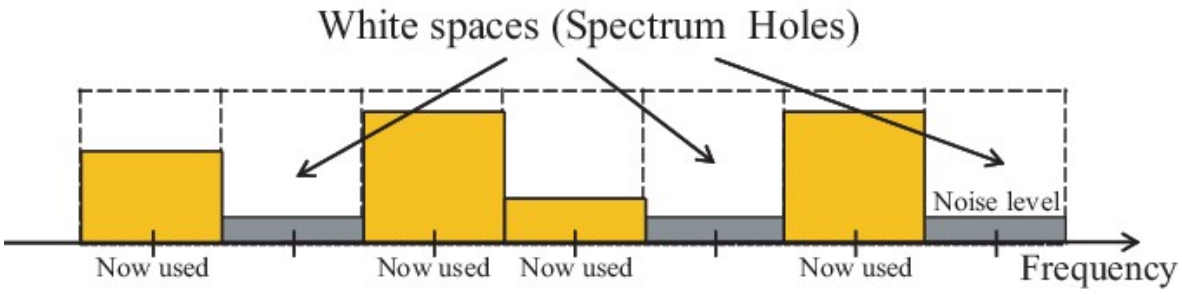


Fig. 1.1: Wideband Spectrum Model

Spectrum Sensing

One of the most critical components of cognitive radio technology is spectrum sensing By sensing and adapting to the environment, a cognitive radio is able to fill in spectrum holes and serve its users without causing harmful interference to the licensed user. One of the great challenges of implementing spectrum sensing is the hidden terminal problem, which occurs when the cognitive radio is shadowed, in severe multipath fading or inside buildings with high penetration loss, while a primary user (PU) is operating in the vicinity. Due to the hidden terminal problem, a cognitive radio may fail to notice the presence of the PU and then will access the licensed channel and cause interference to the licensed system In order to deal with the hidden terminal problem in cognitive radio networks, multiple cognitive users can cooperate to conduct spectrum sensing.

Spectrum Sensing Techniques

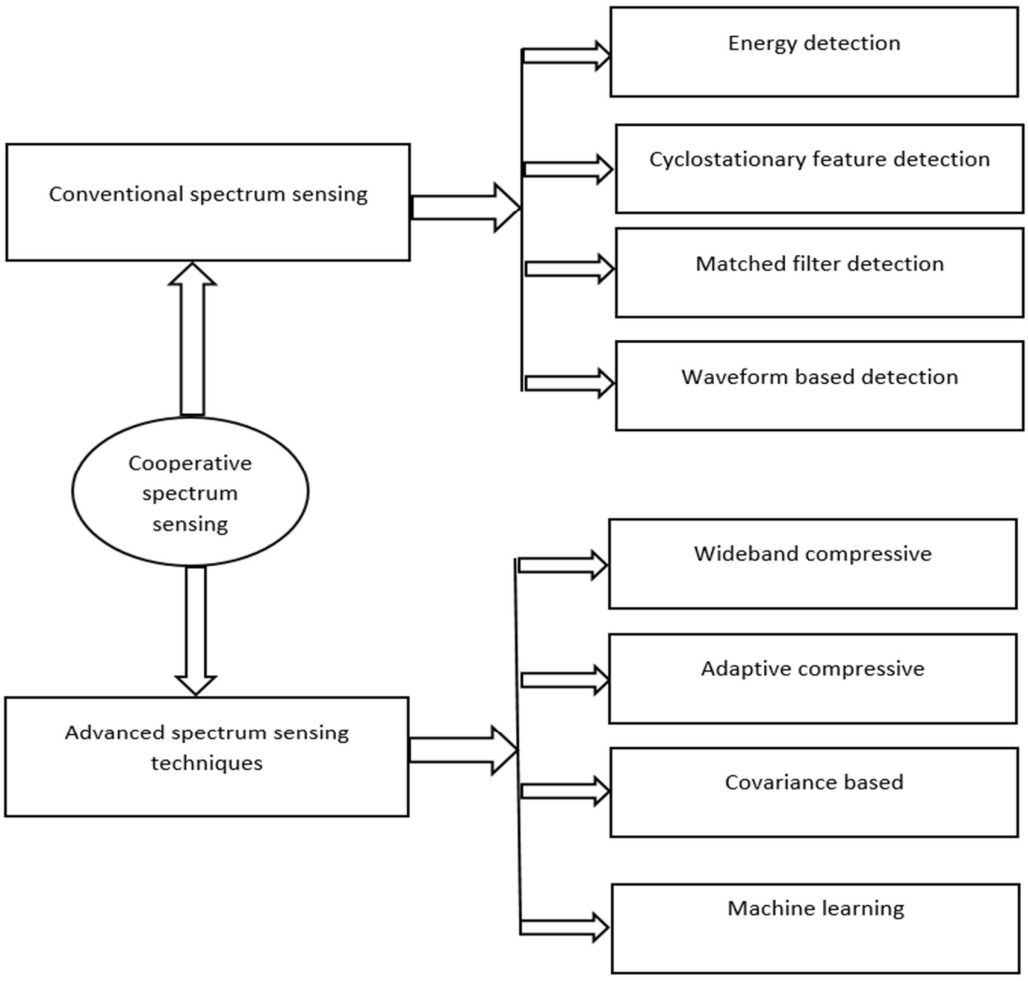


Fig.1.2: Classification of spectrum sensing techniques

Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect. Machine learning is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine leaming has become a significant competitive differentiator for many companies. Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are three basic approaches: supervised learning, unsupervised learning and reinforcement learning.

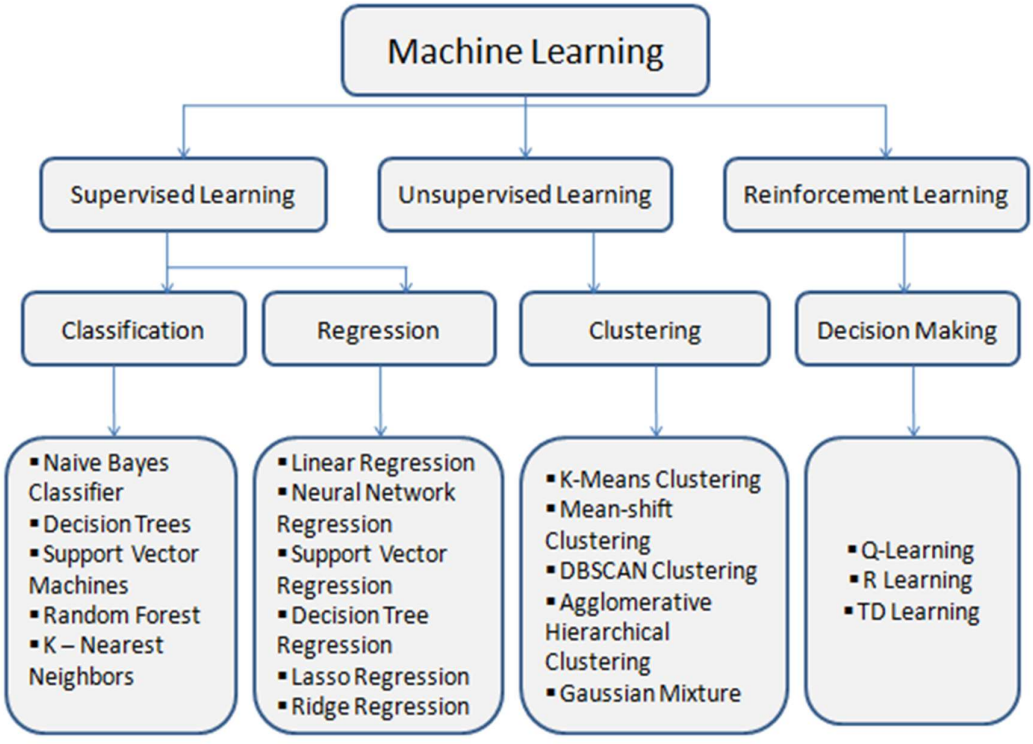


Fig.1.3: Classification of Machine Learning Algorithms

1.2 Aim of the project

The main aim of this project is to sense the wireless communication channel occupancy and to find whether channel is free or not. If the channel is free then it means primary user (PU) is not utilizing the channel then secondary user (SU) can utilize the channel and performance of the communication system in the presence of noise can be improved by utilizing the scarcely available spectrum.

1.3 Methodology

A machine learning-based model can perform better than traditional methods. To identify the existence of a licensed user, we suggest a machine learning based spectrum sensing technique employing a Convolutional Neural Network model in a Gaussian environment. Such that if the spectrum is not being utilized by the licensed user then the secondary user can benefit from it. This methodology will be very helpful in resolving the spectrum scarcity problem in wireless communications.

1.4 Objectives

Presently, due to increase in the number of users in wireless networks against limited radio spectrum, problems have arisen in spectrum scarcity. The conventional allocation of the spectrum is based on proper utilization of the spectrum. However, it is inflexible because each wireless operator has to be given a license to operate at a certain frequency. To overcome this problem, it has become necessary to find out technological means for improved utilization of the spectrum to create opportunities for dynamic spectrum access.

1.5 Outline

In this project report First chapter explains about Introduction to Cognitive Radio, aim of the project, methodology and objectives. Second chapter deals with literature survey, which explains about existing systems and proposed system. Third chapter explains about work title explanation which includes block diagram. Fourth chapter explains about Software details. Fifth chapter explains about results of the project. Sixth chapter explains about Applications, advantages. Seventh chapter explains about Conclusion.

CHAPTER-2

2.1 Literature Survey:

This chapter provides a literature review on the innovations and challenges of the Cognitive Radio technology. Literature survey is the study of already established systems and collection of information which helps in doing new tasks.

K. Davaslioglu, S. Soltani, T. Erpek, and Y. E. Sagduyu [1] proposed a Deep Wi-Fi: Cognitive Wi-Fi with deep learning and discussed about A deep learning-based auto encoder to extract spectrum-representative features and a deep neural network is trained to classify waveforms reliably as idle, Wi-Fi, or jammer and they found that FNN achieves the average accuracy of 98.75 percent in predicting the correct signal labels.

C. Liu, J. Wang, X. Liu, and Y.-C. Liang [2] proposed a Deep CM-CNN for spectrum sensing in cognitive radio and discussed about studied about multi-antenna spectrum sensing problem using DL technology and they found that when PFA = 1.9%, the CM-CNN method could achieve a PD of 96.7%. To overcome the problem of the noise uncertainty, the totally-blind methods have been proposed, which include the blindly combined energy detection, covariance absolute value (CAV) detection.

W. Lee, M. Kim, and D.-H. Cho [3] proposed a deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks and discussed about CNN-based CSS scheme for CRN was proposed, which is the first attempt to use deep learning for CSS and they found that It is observed that the sensing error can be decreased by 17% through the permutation of SU index. The drawback is DCS with small sized CNN structure outperforms conventional CSS schemes especially under harsh sensing conditions which need to be improved.

S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang [4] Spectrum sensing based on deep learning classification for cognitive radio and discussed about Deep convolutional neural network (CNN).CNN-based collaborative sensing method and they found that considered spectrum sensing as a classification problem with two categories and proposed a spectrum sensing method based on deep learning. Classification accuracy rate on the entire test data set is 90.55%. Further the performance of this method can be further improved by transfer learning.

T. J. O'Shea, J. Corgan, and T. C. Clancy [5] proposed a Convolutional radio modulation recognition networks and discussed about the radio using Convolutional Neural Networks (CNNs)

and Deep Neural Networks (DNNs) and they found that Demonstrates that compared to a relatively well expert regarded approach, blind Convolutional Networks on time series radio signal data are viable and work quite well. We achieve roughly a 87.4% classification accuracy across all signal to noise ratios on the test dataset.

Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav [6] proposed a Automatic modulation classification based on constellation density using deep learning and discussed about A constellation density matrix (CDM) based modulation classification algorithm is proposed to identify different orders of ASK, PSK, and QAM And they found that Parameters of the solver configuration have also been adjusted for better classification and fast training processes, such as the learning rate is 0.0001, and the mini-batch size is 64. For training, −4 dB to 30 dB SNR with the step of 1 dB and 35 dB to 70 dB SNR.

Captain, Kamal M., and Manjunath V. Joshi [7] proposed a Spectrum Sensing for Cognitive Radio: Fundamentals and Applications and discussed about The most widely used spectrum sensing methods are energy detection and matched filter detection and they found that ROC curves are generated by plotting either detection probability versus false alarm probability or missed detection probability versus false alarm probability. Each value is assigned probability between 0 and 1, and the sum of all probabilities is equal to 1.For bell-shaped curve symmetric around μ mean 0 and variance 1, i.e., $\mu = 0$ and $\sigma 2 = 1$. In future studies, the aim is to apply and verify the performance of the proposed algorithm on different spectrum sensing methods and also focus on the optimization of some expressions used in the algorithm to reduce mathematical complexity and improve detection time.

Xie, Jiandong [8] proposed a Deep learning-based spectrum sensing in cognitive radio: A CNN LSTM approach and discussed about the research of spectrum sensing is focused on deep learning which is free from model assumptions and they found that the CNN-LSTM structure is free of the signal-noise model assumptions and moreover, To overcome the problem of the noise uncertainty, the totally-blind methods have been proposed, which include the blindly combined energy detection, covariance absolute value (CAV) detection. It is able to simultaneously learn the signal energy-correlation features and the temporal PU activity pattern features to promote the detection performance. The false alarm probability is set to $Pfa = 0.1$.

2.2 Early Developments:

Researchers at the University of California at Berkeley have proposed an experimental setup based on the Berkeley Emulation Engine 2 (BEE2) platform to compare different sensing techniques and develop metrics and test cases so as to measure the sensing performance. A distributed genetic algorithm based CR engine is proposed in the center for wireless telecommunications at Virginia Tech. The cognitive engine focuses on how to provide CR capability to the physical and MAC data link layers. Researchers at Rutgers University have constructed an Open Access Research Testbed for Next- Generation Wireless Networks (ORBIT). IEEE 802.22 proposed to reuse the fallow TV spectrum without harmful interference to TV incumbents. A CR based PHY and MAC for dynamic spectrum sharing of vacant TV channels is evaluated. Dynamic frequency hopping (DFH) is recently proposed in IEEE 802.22, where sensing is performed on the intended next working channels in parallel to data transmission in current working channel and no interruption is required for sensing. The IEEE P1900 is a new standard series focusing on next generation radio and spectrum management. One important focus of the standard is to provide reconfigurable networks and terminals in a heterogeneous wireless environment.

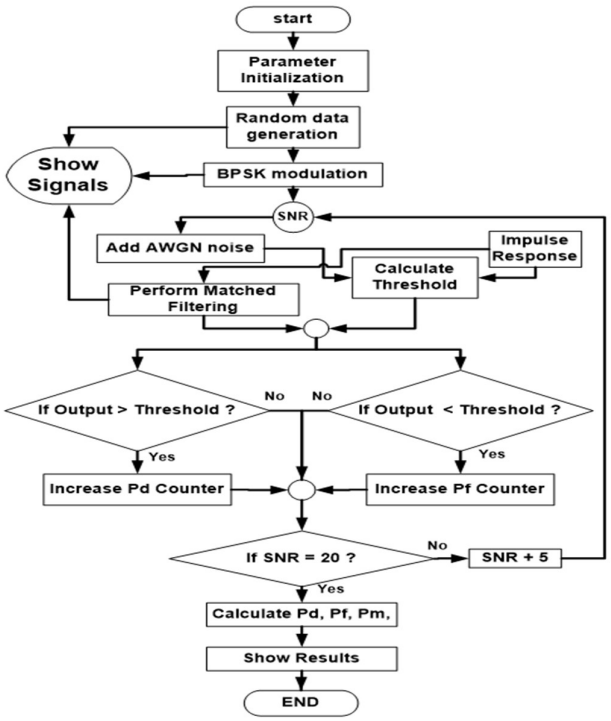


Fig. 2.2: Flowchart based on Matched Filter Detection

CHAPTER-3

3.1 Spectrum Sensing

The 'Cognitive Radio' (CR) technology is the best technology available to improve the spectrum utilization in wireless communications. The CR is used to provide reliable communication between the users at any time by effectively utilizing the spectrum. The key function of the CR is the spectrum sensing that opportunistically detects the unutilized spectrum, which is licensed to the primary user (PU). Spectrum sensing is an important functionality of CR to improve the spectrum utilization and at the same time limiting the harmful interference to the licensed users.

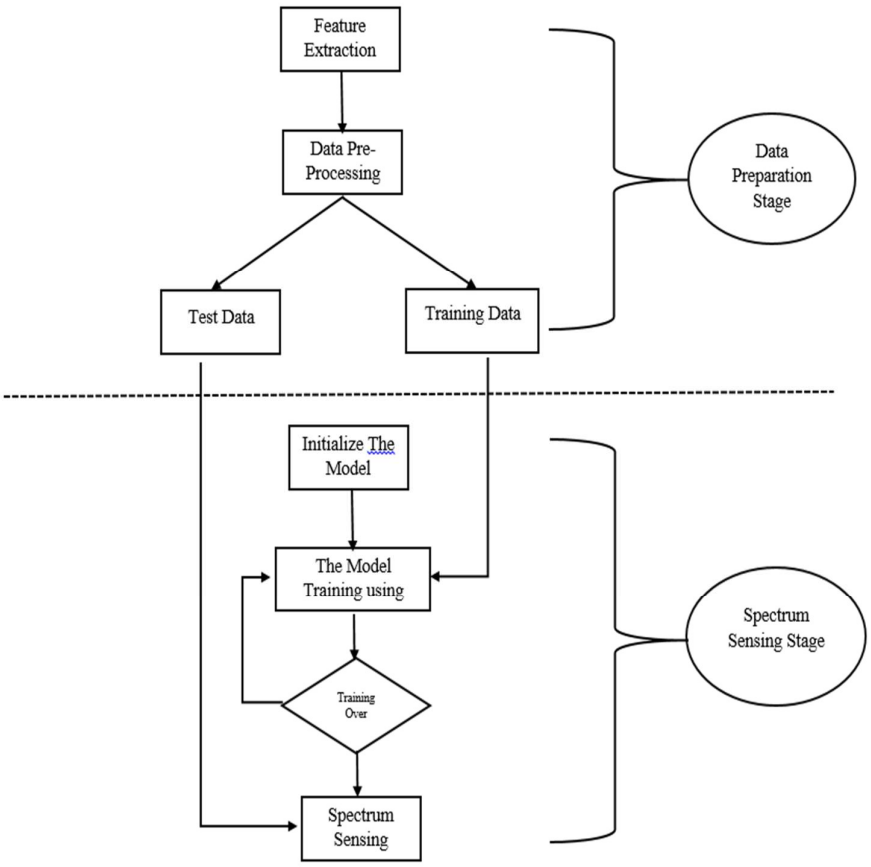


Fig. 3.1: Flow chart of cooperative spectrum sensing

Feature Extraction:

Feature extraction involves selecting and transforming the relevant data from the dataset into a set of features that can be used to train a machine learning model. This process may include tasks such

as data cleaning, data normalization, and feature engineering. Data cleaning involves removing any noise in the data that may negatively affect the performance of the machine learning model. Data normalization involves scaling the data to a standard range. Feature engineering involves creating new features from the existing data to improve the performance of the machine learning model. Overall it helps to ensure that the data is in a suitable format for the machine learning.

Data Pre-Processing:

Data preprocessing may involve several steps, including

1. Data cleaning: This step involves identifying and handling missing or corrupted data Techniques like imputation or deletion can be used to handle missing data.

2. Data transformation: This step involves transforming the data to ensure that it meets the assumptions of the machine learning model. Techniques like scaling, normalization, or log transformations can be used to transform the data.

3. Feature selection: This step involves selecting the most relevant features that are important for the machine learning model Feature selection can help to reduce the dimensionality of the dataset and improve the accuracy of the model.

4. Feature engineering: This step involves creating new features from the existing data to improve the performance of the machine learning model Feature engineering can help to capture nonlinear relationships in the data and improve the accuracy of the model.

5. Data splitting: This step involves splitting the data into training and testing sets. The training set is used to train the machine learning model while the testing set is used to evaluate the performance of the model. Overall, data preprocessing is an important step in the data preparation stage of a machine learning project. It helps to ensure that the data is in a suitable format for the machine learning algorithm to learn from and can significantly improve the accuracy of the model.

Training data:

Training data is used to train a machine learning model. It is the set of data that is fed into the model during the training process. The model learns from this data, and its parameters are adjusted to minimize the error between its predictions and the true labels of the training data.

Testing data:

Testing data is used to evaluate the performance of the trained machine learning model. It is a set

of data that model has not seen before. The model uses the features of the testing data to make predictions, and its performance is evaluated by comparing its predictions to the true labels of the testing data.

Spectrum sensing stage:

Spectrum sensing is a crucial stage in cognitive radio networks, where the goal is to detect the presence or absence of primary user's signal in a given frequency band. The spectrum sensing stage is the first step in enabling cognitive radio networks to utilize unused spectrum bands while avoiding interference with primary users.

3.2 METHODOLOGY

Here we are implemented this model by using Convolution Neural Networks Algorithm. The CNN is as a prominent deep architecture of deep learning. CNN includes multiple layers of representations. Due to this deep structure, CNN can automatically obtain the representation characteristic from the raw data through nonlinear transformations and approximate nonlinear functions. A typical CNN structure consists of a feature extractor which is composed of several convolutional layers usually followed by pooling layers and a softmax classifier. The convolutional layer extracts signal features, whereas the pooling layer reduces the dimensions and thus further reduces the computation time. This architecture can attain a form of regularization by itself. The features extracted are then put into the top softmax layer for classification. CNN has shown its powerful ability to extract information and useful features. The signals are first processed by the CNN to obtain the features, which are extracted from the last hidden layer of the CNN. The performances of the CNN and SVR have improved in rotating machine fault detection when they were combined.

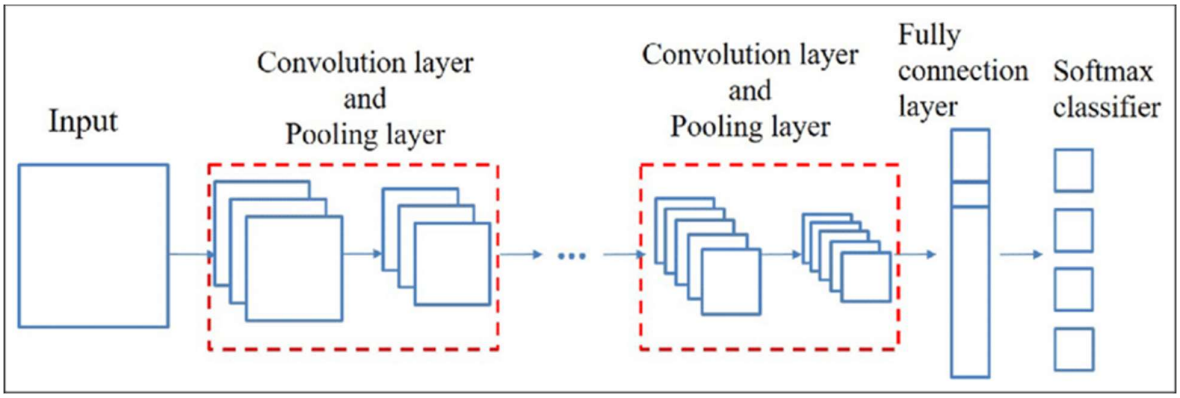


Fig. 3.2: Architecture of CNN

Convolution Layers

A 2D convolution layer with 256 filters is a type of layer commonly used in deep learning neural networks for image recognition and processing tasks. In simple terms, a 2D convolution layer applies a mathematical operation to each small region of an input image, called a kernel or filter, to extract features from the image. The layer has 256 filters, which means that 256 different feature maps are generated from the input image, with each filter detecting a specific type of feature.

Pooling Layer

An average pooling 2D layer is a type of layer in convolutional neural networks that reduces the spatial dimensions of the input volume by taking the average of adjacent values in each channel. In simple terms, the layer divides the input image into a set of non-overlapping rectangular regions, or "pools", and replaces the values in each pool with their average. This process effectively reduces the size of the image while retaining the most important information. For example, if the input image has a size of 28x28x64 (width, height, and depth), the average pooling layer may divide the image into 2x2 pools, resulting in an output volume of size 14x14x64.

Flatten Layer

In a Convolutional Neural Network (CNN), the Flatten layer is a layer that is typically used to convert the output from the previous convolutional layer(s) into a 1-dimensional feature vector that can be fed into a fully connected layer. The Flatten layer takes the 3-dimensional output tensor (height, width, channels) from the previous convolutional layer and transforms it into a 1-dimensional array.

Gaussian Noise 2D layer

A Gaussian noise 2D layer is a layer that adds 2D Gaussian noise to the input data in a neural network. This type of layer is commonly used in deep learning models to add noise to the input data, which can help to prevent overfitting and improve the generalization performance of the model.

The 2D Gaussian noise layer works by generating a random matrix of the same shape as the input data, with values drawn from a Gaussian distribution with mean 0 and standard deviation specified by a hyperparameter. The random matrix is then added element-wise to the input data.

The effect of adding Gaussian noise is to introduce random variations into the input data, which can help to prevent the model from overfitting to the training data. This is because the added noise effectively expands the size of the training set, making it more difficult for the model to memorize the training data. In addition to preventing overfitting, adding Gaussian noise can also have a regularization effect, helping to smooth the decision boundary of the model and improve its generalization performance on new, unseen data.

In summary, a Gaussian noise 2D layer adds random 2D Gaussian noise to the input data in a neural network, which can help to prevent overfitting and improve the generalization performance of the model.

Dense layers

In a Convolutional Neural Network (CNN), a Dense layer with 1024 units is a fully connected layer with 1024 neurons. The purpose of the Dense layer is to take the output of the previous convolutional layers and transform it into a format that can be fed into the final output layer of the CNN. The output of the last convolutional layer is typically a 3-dimensional tensor, with dimensions (height, width, channels). Before being passed to the Dense layer, this tensor is typically flattened into a 1-dimensional feature vector using a Flatten layer.

In summary, a Dense layer with 1024 units in a CNN takes a flattened feature vector as input and applies a set of learnable weights and biases to compute a linear transformation of the input features. The result is then passed through a non-linear activation function to learn a set of higherlevel features from the input data.

Softmax layer

In a Convolutional Neural Network (CNN), the softmax function is often used as the final activation function in the output layer. The purpose of the softmax function is to convert the output of the final Dense layer (which typically contains a set of real-valued scores) into a probability distribution over the possible output classes.

The softmax function takes a vector of real-valued scores as input and applies a mathematical operation to normalize the scores so that they sum to 1. In the context of a CNN, the output of the final Dense layer represents the confidence of the network in each possible output class. During inference, the softmax function is used to make predictions by selecting the class with the highest probability.

3.3 PROPOSED SYSTEM

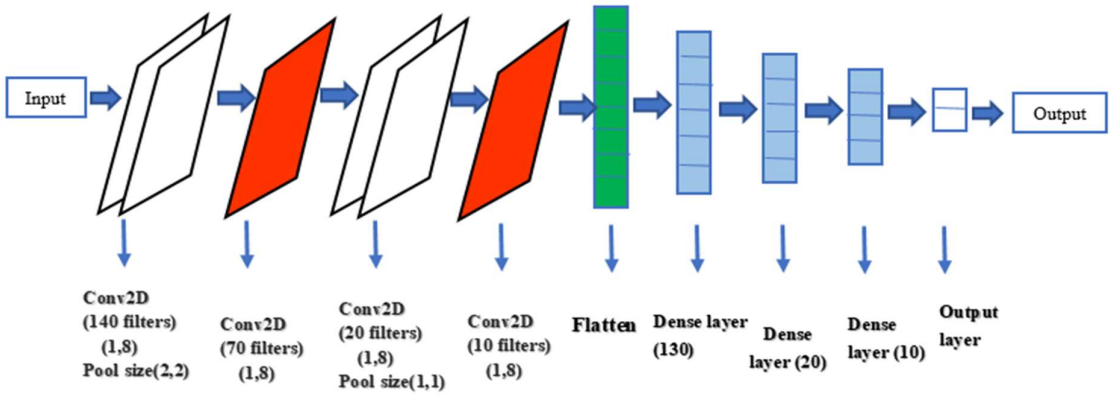


Fig. 3.3: Proposed diagram based on CNN

The feed forward, involves supplying the model with data in order to obtain the layer-by-layer output. Then, we use an error function to calculate the error. Then, we back produce the model after computing variables. In general Back propagation is the process that is used to reduce waste. In this article swish activation function with lecun uniform and lecun normal kernel initializers have been proposed, and in the final stage for classification purpose sigmoid activation has been used with adadelta optimizer.

The Swish activation function, has been shown to outperform other popular activation functions such as ReLU and sigmoid in various deep learning tasks. Swish is defined as the product of the input and the sigmoid function of the input, or mathematically:

Swish(x) = x* sigmoid(beta*x)

where beta is a parameter that controls the slope of the function.

Lecun uniform and Lecun normal kernel initializers are weight initialization methods. These initialization methods are specifically designed to work well with activation functions that have a sigmoid-like shape, such as the Swish function Lecun uniform and Lecun normal initializers set the initial weights of the network using a random distribution with a mean of zero and a standard deviation that depends on the number of input and output neurons

For the final stage of classification, the sigmoid activation function is commonly used to output probabilities of the different classes. The Adadelta optimizer is an adaptive learning rate optimizer that is well suited for deep neural networks, as it automatically adjusts the learning rate based on the gradient of the loss function.

The above diagram fig.1 illustrates the working of the proposed model. The data set is the input to the convolution layer, which typically represents the pixel matrix of the data. The input layer is fed to 2-dimensional convolutional layer with 140 filters, each of which has size of 1x8

(meaning of filter size is 1 along the vertical dimension and 8 along the horizontal dimension). The pool size (2, 2) indicates that after the convolutional operation, a max-pooling operation with a pool size of 2x2 is applied to the output layer of the convolutional layer.

Overall, this layer is meant to glean 140 sets of feature sets from the input data, each of which corresponds to a different filter, and then reduce the spatial size of the feature maps through max-pooling. The next layer is the 2 dimensional convolutional layer with 70 filters each of which has size 1x8. This layer is designed to extract 70 sets of features from the input data each of which corresponds to a different filter. The spatial size of the output feature maps from the convolutional layer is already small enough. So there is no need to apply max-pooling to further reduce the size. The next layer is Conv2D which consists of 20 kernels which has size 1x8.

The pool size of 1x1 is fed to decision layer of feature extractor. Actually this type of pooling operation doesn't actually perform any pooling or down sampling of the input data. Instead, it simply passes the input data through unchanged. This is fed to Conv2D with 10 kernels which has size 1x8 and there is no max-pooling layer because the size is already reduced. The layer next to the 2 dimensional Convolutional layer is the flatten layer. A conv2d layer is used to extract features and generate 2D feature maps as output. In order to pass this information to fully connected layer that expects 1D input, the feature maps need to be flattened into a vector. It is fed to a dense layer with 130 neurons, it means that the flattened vector is being treated as a 1D input to a fully connected layer with 130 neurons. The flattened vector contains the feature information that has been extracted by the previous layers in the network.

For developing this architecture model in this article deepsig dataset that is radioml 2016.04.C has been considered and was downloaded from the following RF datasets for Machine Learning website: https://www.deepsig.ai/datasets. Overall the result of the network's computation on the input data and can be used for a variety of tasks, including prediction, classification, and regression.

Historical Dataset: RADIOML 2016.04C

This is a variable-SNR dataset with moderate LO drift, light fading, and numerous different labeled SNR increments for use in measuring performance across different signal and noise power scenarios. A machine learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions. DeepSig has created a small corpus of standard datasets which can be used for original and reproducible research, experimentation, measurement and comparison by fellow scientists.

CHAPTER-4
SOFTWARE DESCRIPTION

4.1 Introduction to Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. It has fewer syntactical constructions than other languages.

- Python is interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- Python is a Beginner's Language: Python is a great language for the beginner level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features:

Python's features include-

- Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows a student to pick up the language quickly.

- Easy-to-read: Python code is more clearly defined and visible to the eyes.

- Easy-to-maintain: Python's source code is fairly easy-to-maintain.

- A broad standard library: Python's bulk of the library is very portable and crossplatform compatible on UNIX, Windows, and Macintosh.

- Interactive Mode: Python has support for an interactive mode, which allows interactive testing and debugging of snippets of code.

• Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

• Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

• Databases: Python provides interfaces to all major commercial databases.

• GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

• Scalable: Python provides a better structure and support for large programs than shell scripting.

4.2 Libraries/Packages

Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Seaborn:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn library aims to make a more attractive visualization of the central part of understanding and exploring data. It is built on the core of the matplotlib library and also provides dataset-oriented APIs.

Seaborn is also closely integrated with the Panda's data structures, and with this, we can easily jump between the various different visual representations for a given variable to better understand the provided dataset.

Sklearn:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. Scikit-learn, often abbreviated as "sklearn", is a popular open-source Python machine learning library that provides a wide range of tools and functionality for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and evaluation. Scikit-learn is built on top of other Python libraries such as NumPy, SciPy, and matplotlib, and provides a consistent and user-friendly interface for performing machine learning tasks efficiently and effectively.

Keras:

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers,

Francois Chollet. It is made user-friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination. Keras was designed to be modular and easy to use, allowing researchers and developers to quickly build and experiment with various types of deep learning models. It provides a user-friendly interface for designing, training, and evaluating deep learning models.

Pickle:

Pickle in Python is primarily used in serializing and deserializing a python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. The pickled byte stream can be used to re-create the original object hierarchy by unpickling the stream. The pickle package in Python is a module that provides a way to serialize and deserialize Python objects, allowing you to convert complex Python objects into a format that can be saved to a file or transmitted over a network, and then restored back into Python objects when needed. This process is also known as object serialization and deserialization.

GZip:

GZip application is used for compression and decompression of files. It is a part of GNU project. Python's gzip module is the interface to GZip application. The gzip data compression algorithm itself is based on zlib module. The gzip package in Python is a module that provides functionality for compressing and decompressing data using the gzip compression format, which is a popular lossless data compression format used for reducing the size of files or data streams. The gzip format is widely used for compressing text files, such as log files, and is supported by most operating systems and web browsers.

Tensorflow:

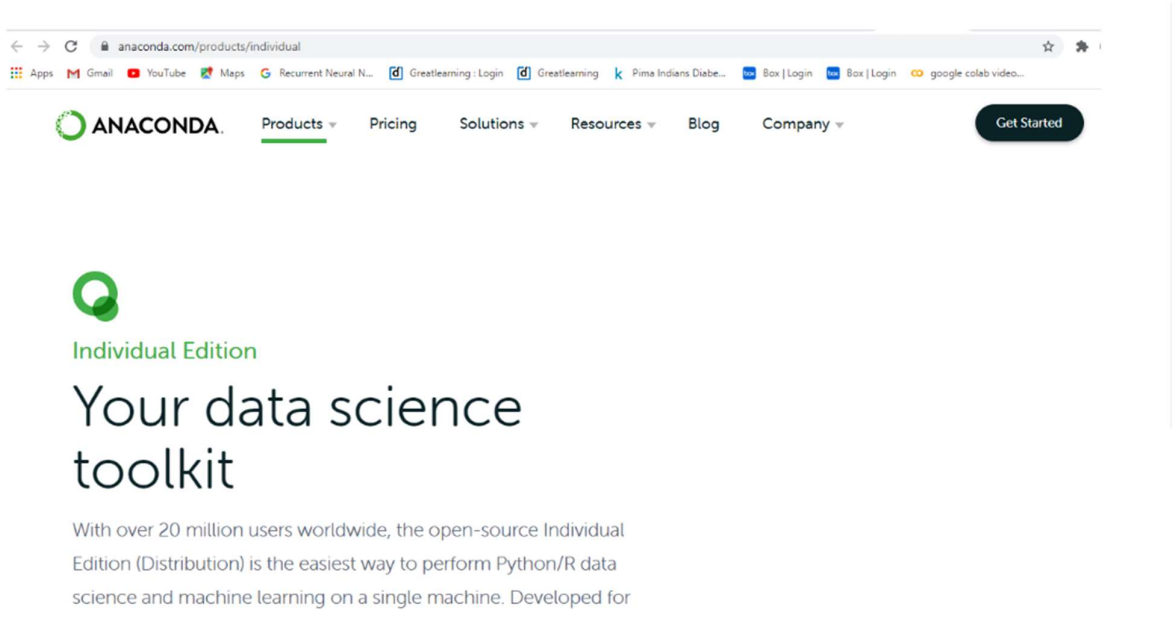
TensorFlow is an open-source machine learning and deep learning framework developed by Google that provides a comprehensive ecosystem of tools, libraries, and resources for building and training machine learning models. It is one of the most popular and widely used frameworks for developing machine learning and deep learning applications in Python. Overall, TensorFlow is a powerful and versatile machine learning and deep learning framework that provides a wide range of tools and resources for building, training, and deploying machine learning models in Python. It is widely used in various industries and research domains for a wide range of applications, including computer vision, natural language processing, speech recognition, recommendation systems, and more.

4.3 Anaconda software installation steps

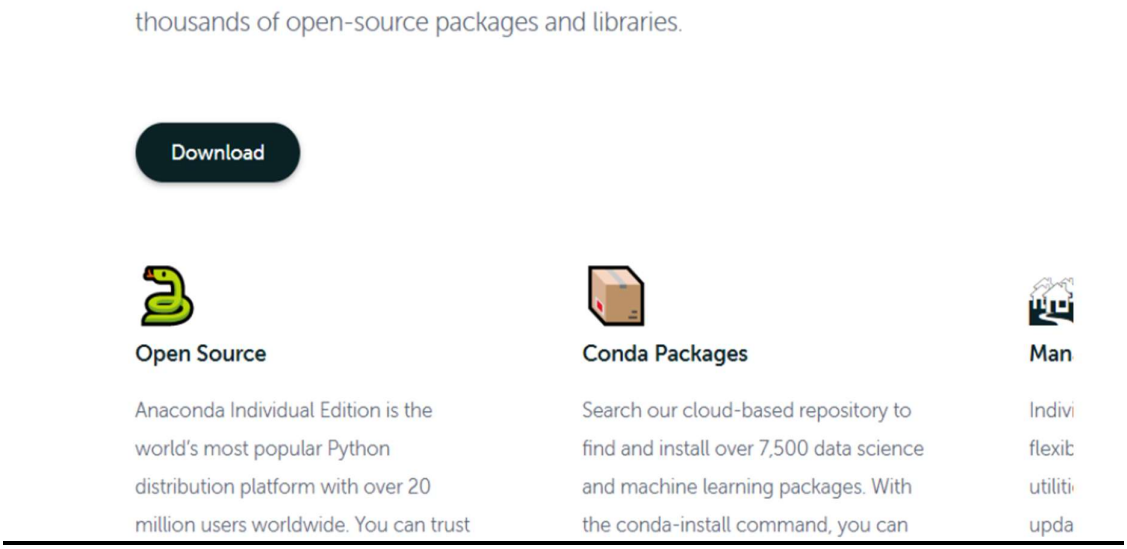
Jupyter Notebook is an open-source web application. This application allows you to create documents that can contain live code, equations, visualizations, images, and narrative text. This application is mainly used for data science or statistical evaluation purpose.

In order to install Jupyter using Anaconda, Please follow the following instructions:

Step 1: Installation of Anaconda

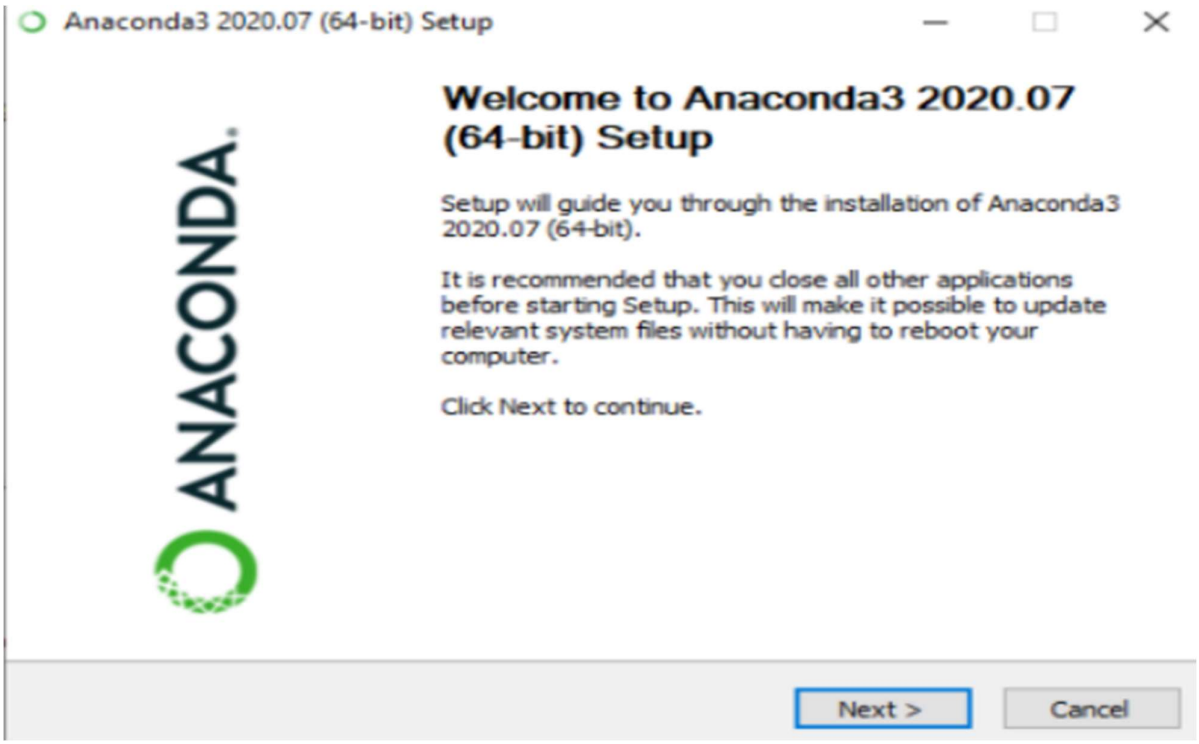


Step 2: *Please go to the* Anaconda.com/downloads **site**



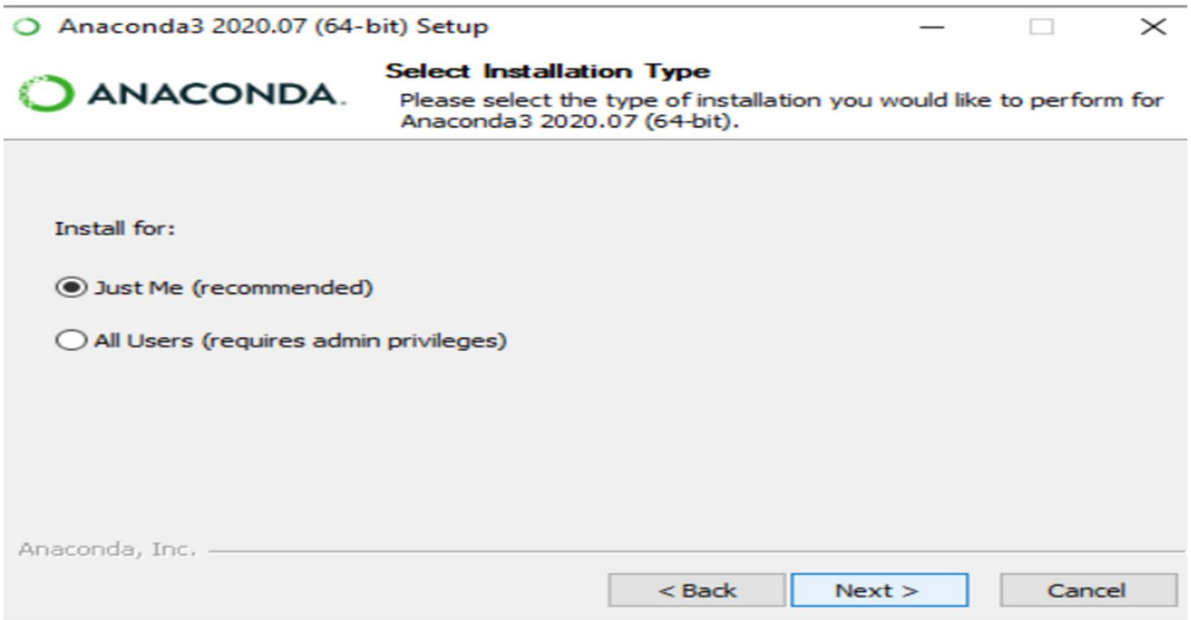
Step 3: Select the respective platform: Windows/Mac/Linux

Step 4: Download the *.exe* installer

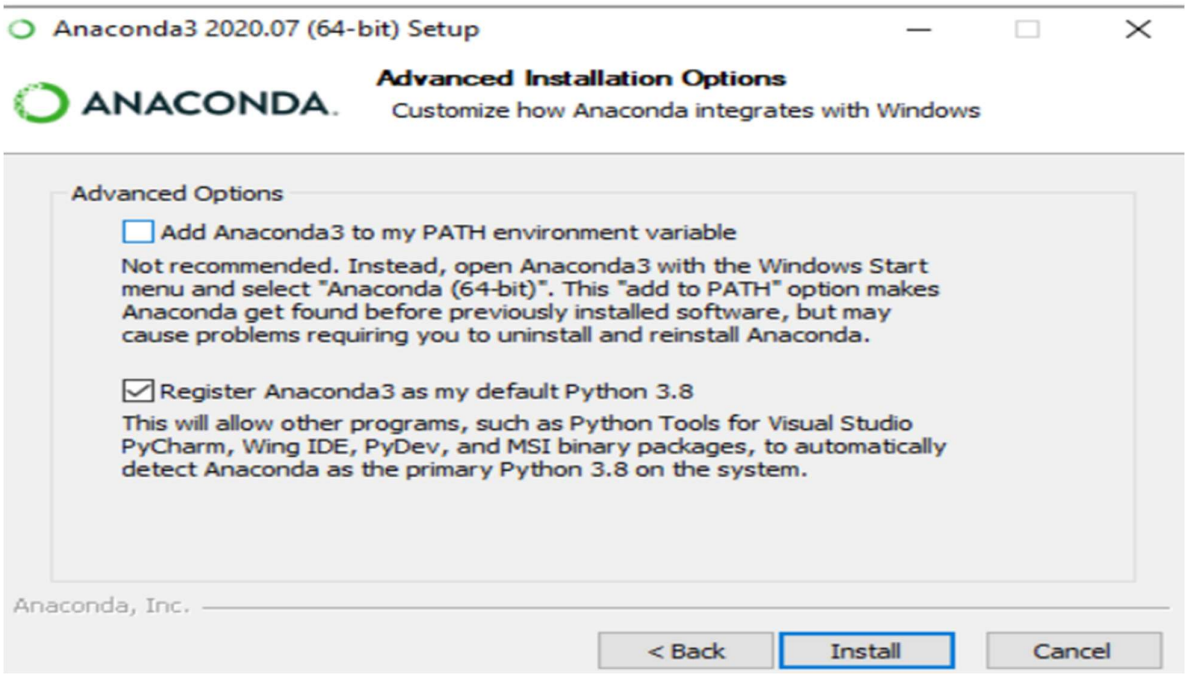


Step 5: Open and execute the *.exe* installer

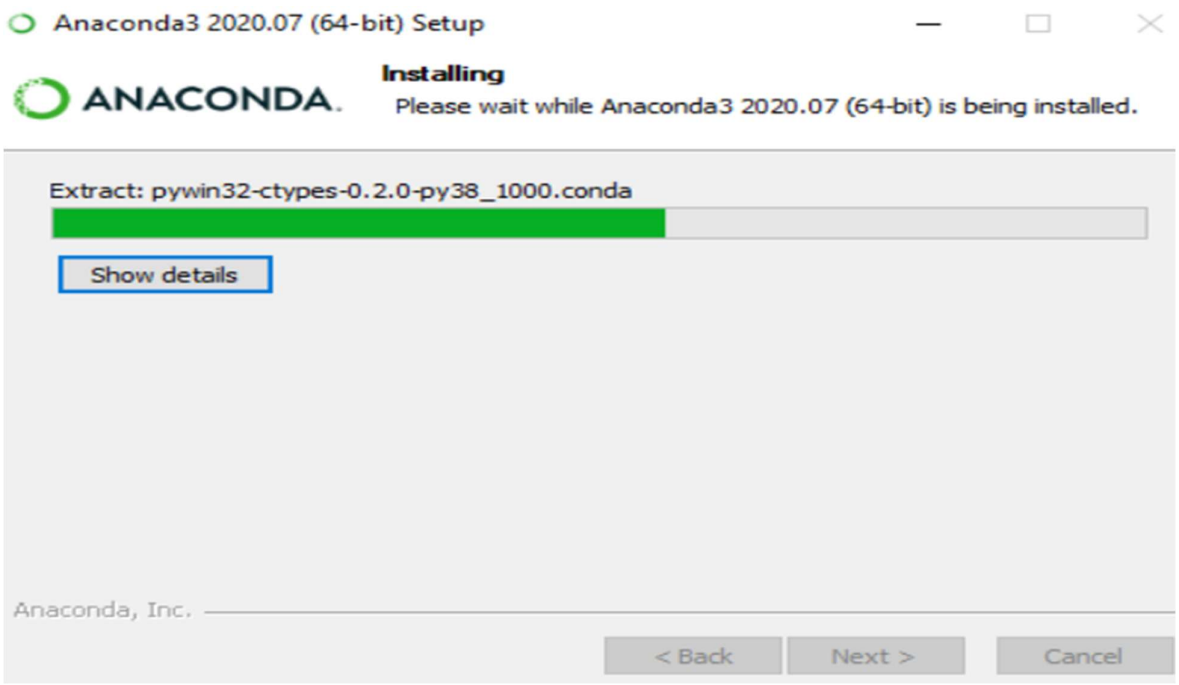
Step 6: Launch Anaconda Navigator



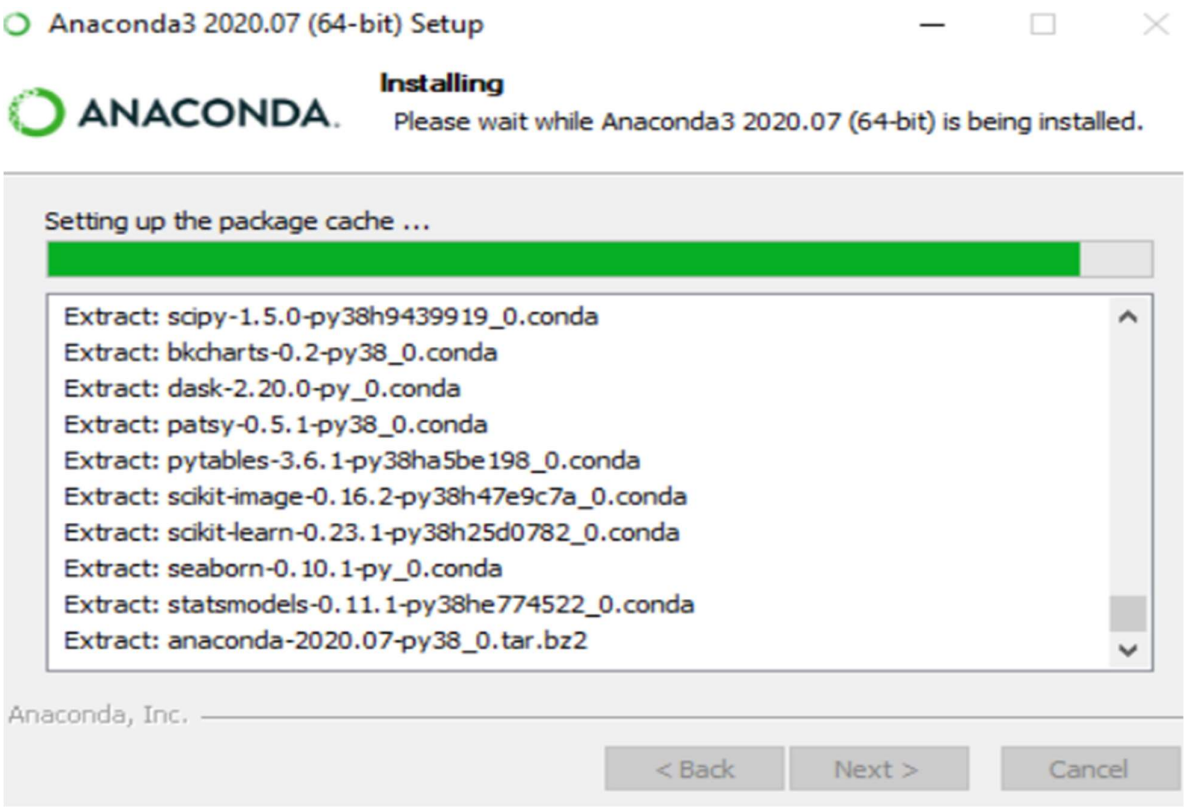
Step 7: Click on the Install Jupyter Notebook Button



Step 8: Beginning the Installation

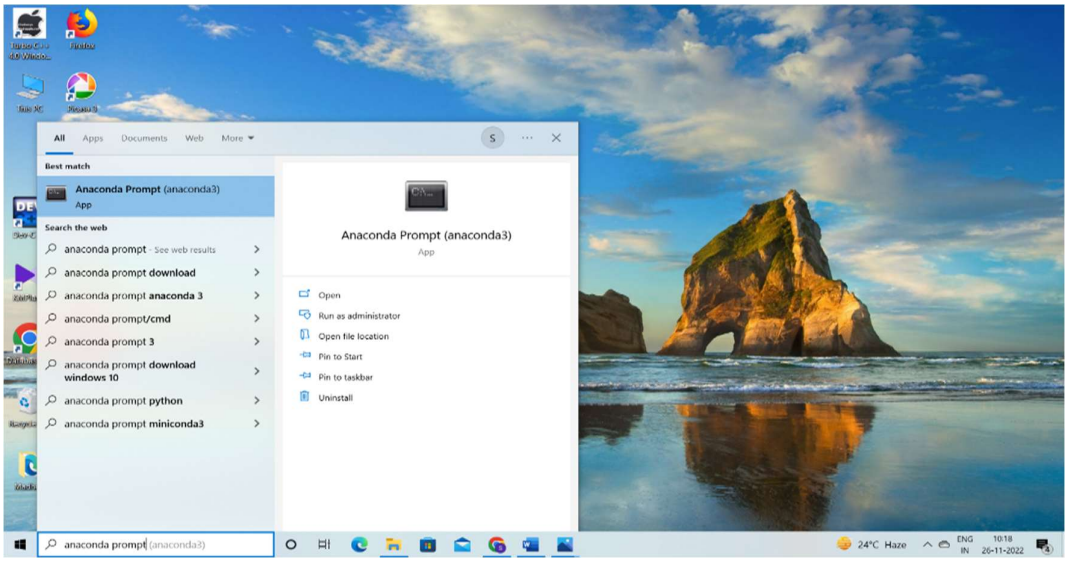


Step 9: **Loading Packages**

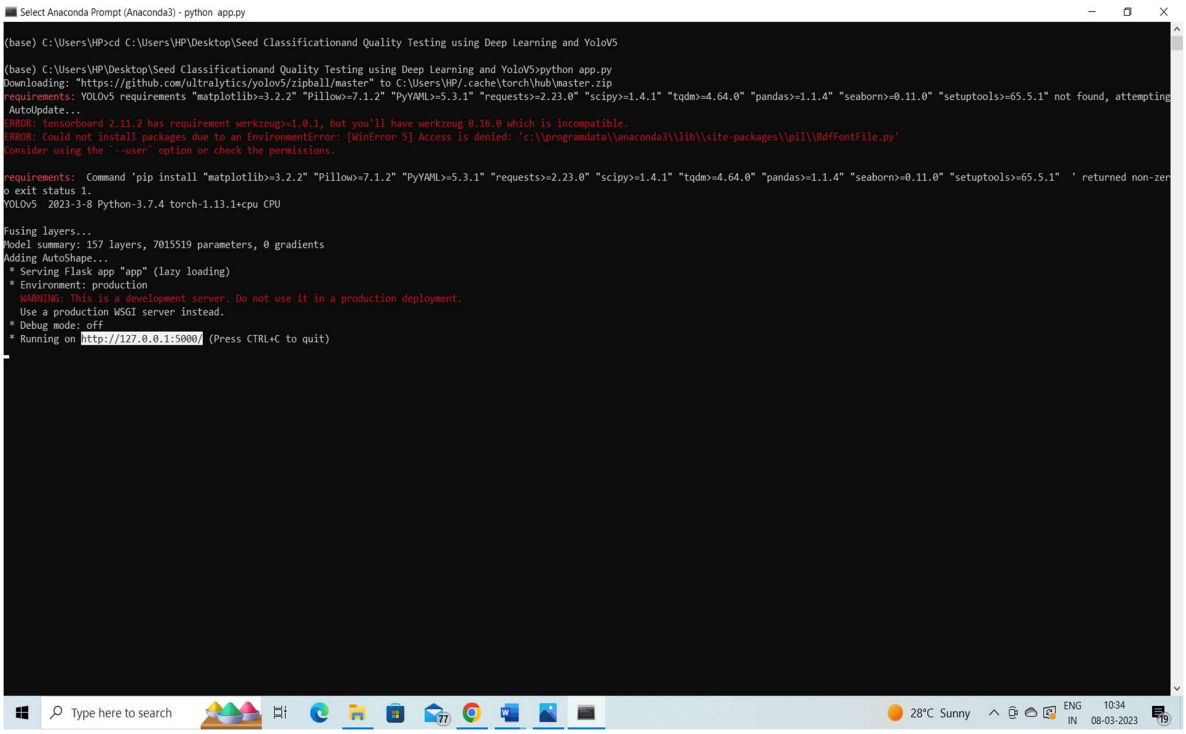


Procedure For Execution

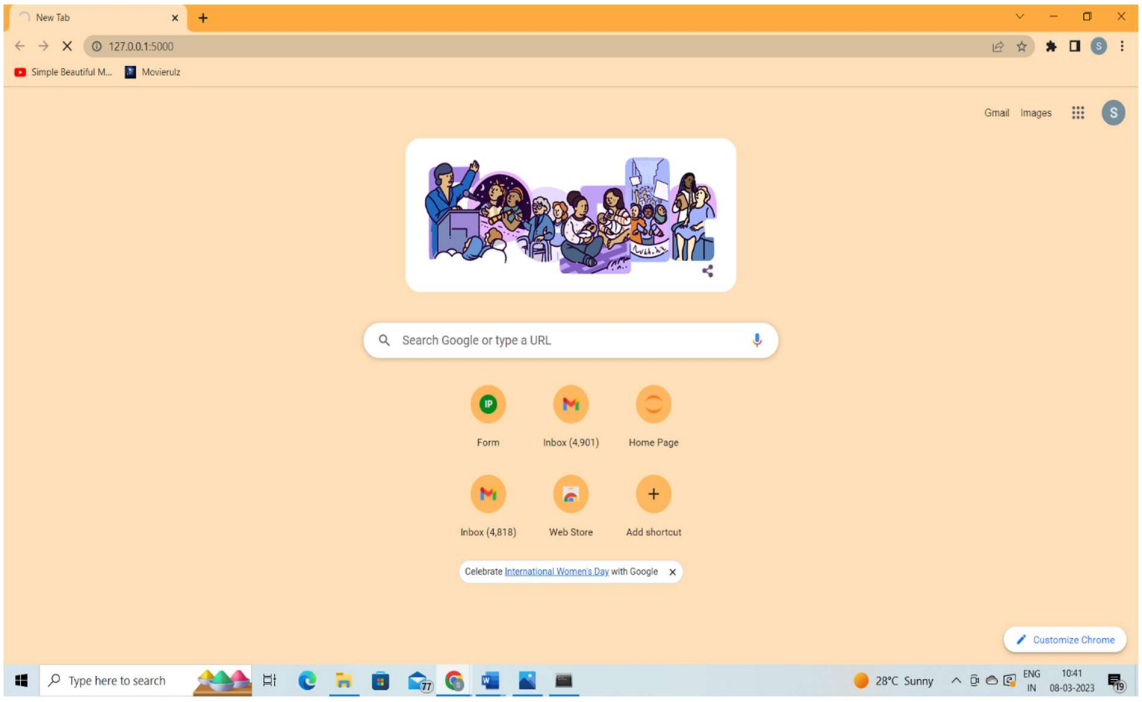
<u>Step 10:</u> Search for anaconda prompt



Step 11: After giving the path copy the URL



Step 12: Paste the URL in the browser

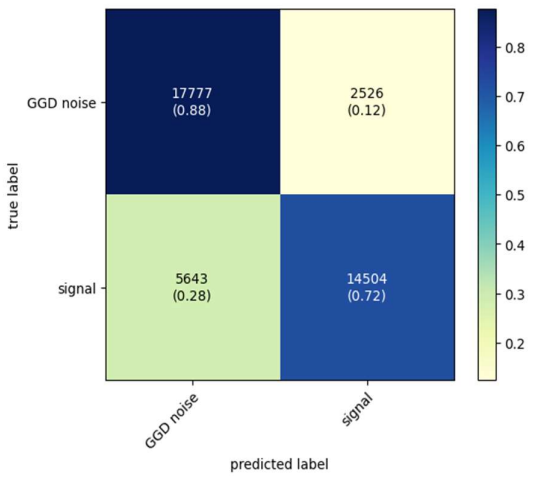


CHAPTER-5

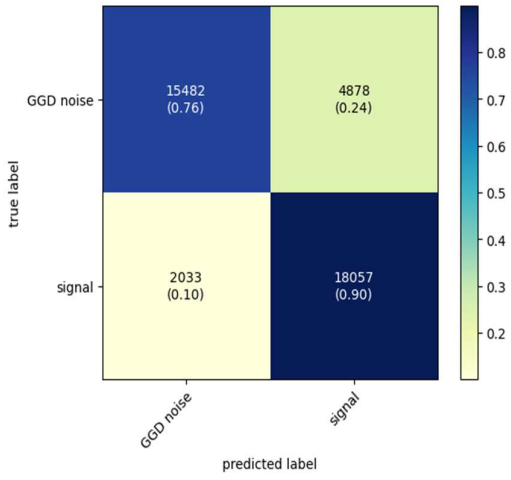
5.1 Results:

In conclusion, confusion matrices are used to assess an identification model's efficiency, and activation functions are utilized for adding non-linearity to a neural network model. While it is unrelated to the process of creating a confusion matrix, the choice of an activation function can have an effect on a model's success. Though it is irrelevant to the method for building a confusion matrix, the selection of an activation function may have an effect on a model's success. Since different activation processes like swish, relu, selu, and gelu have differing confusion matrices, addressing them may not be proper. By careful observation, we can be to clearly identify that the confusion matrix of the activation function SELU has high accuracy when compared with confusion matrix of other activation functions like Swish, ReLU and GELU. The confusion matrix of different activation functions at SNR level of 16 dB is shown below. From the figure shown below we can clearly understand that the classification accuracy is accurate with Selu when compared to other activation functions. The amount of percentage of accuracy increases as the SNR value increases. We have noticed that at low SNR value of -20dB the percentage of the accuracy is low and as the SNR value increases the amount of percentage of accuracy has increased. At SNR value of 16 dB, we observed the percentage of accuracy has increased at good level as we compared it with the accuracy that we have obtained at -16 dB.

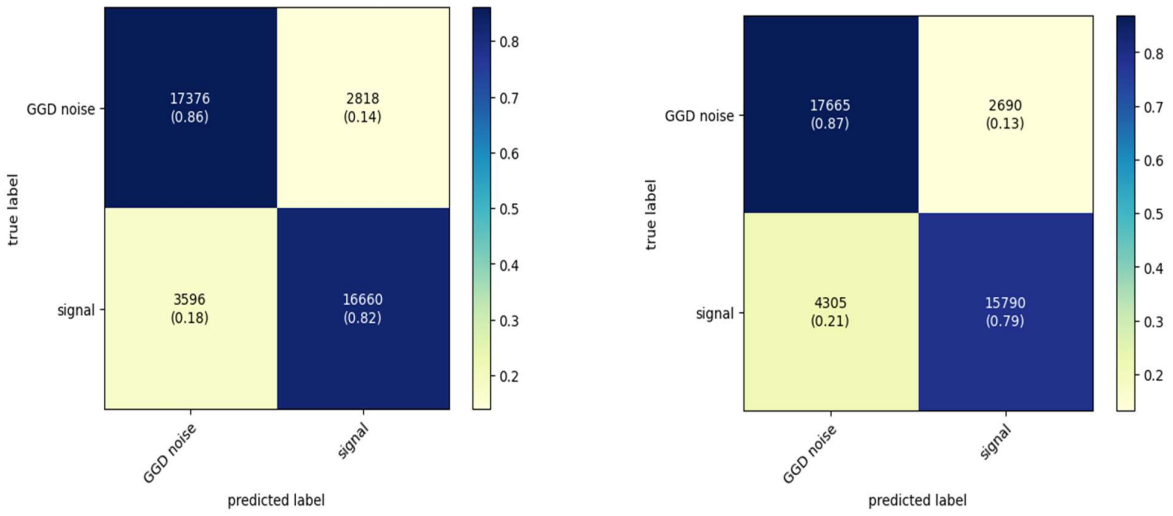
Confusion matrix of True label vs Predicted label



GeLU



ReLU



SeLU **Swish**

Fig. 5.1: Confusion matrix of different activation functions

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix. Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.

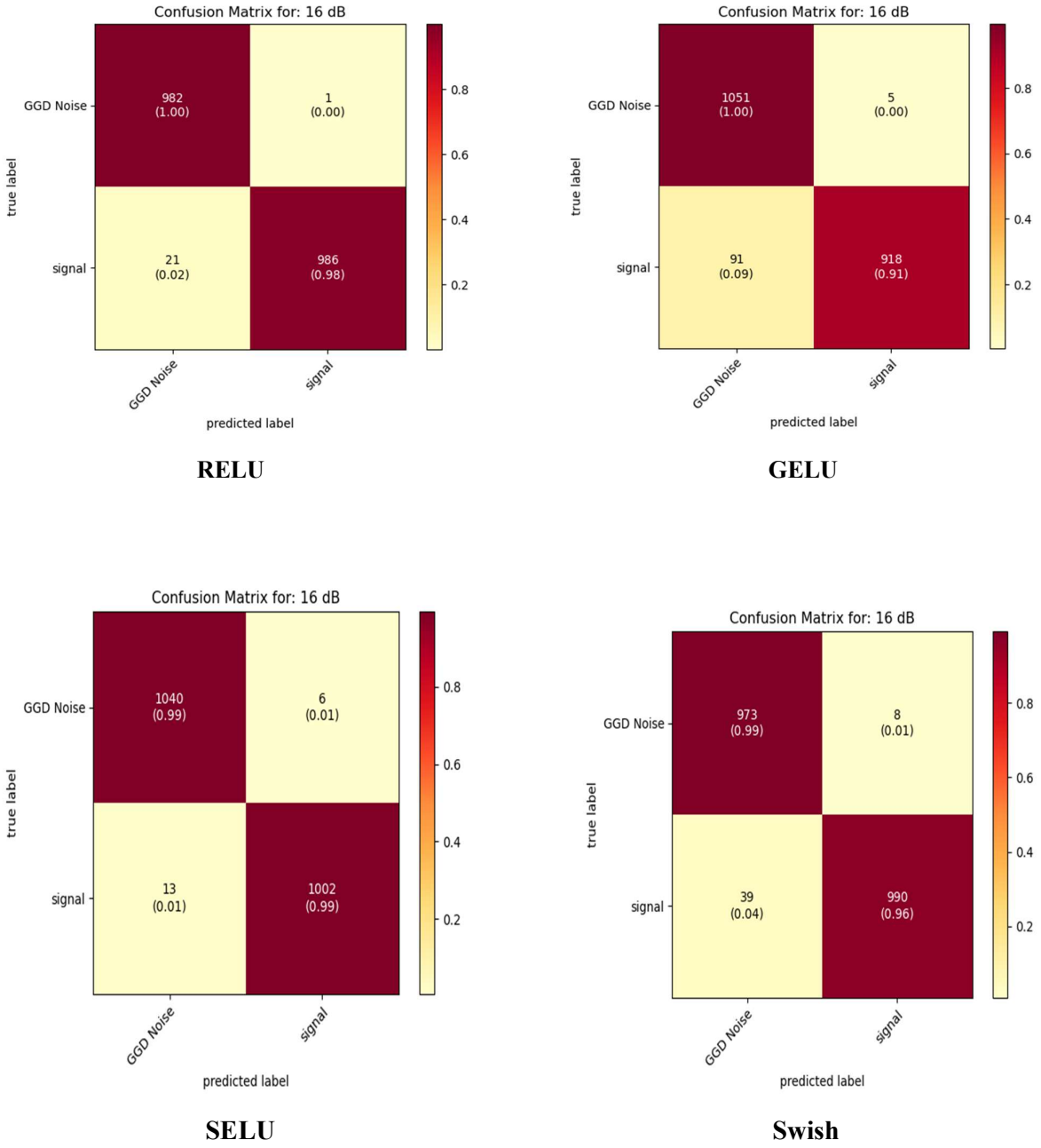
True Negative: Model has given prediction No, and the real or actual value was also No. This refers to the cases where the model correctly predicts the negative class and the true underlying class label is also negative.

True Positive: Model has predicted yes, and the actual value was also true. This refers to the cases where the model correctly predicts the positive class and the true underlying class label is also positive

False Negative: The model has predicted no, but the actual value was Yes, it is also called as Type-II error. This refers to the cases where the model incorrectly predicts the negative class, but the true underlying class label is actually positive

False Positive: The model has predicted Yes, but the actual value was No. It is also called as Type-I error. This refers to the cases where the model incorrectly predicts the positive class, but the true underlying class label is actually negative.

Confusion matrix of different activation functions at SNR 16 dB



RELU **GELU**

SELU **Swish**

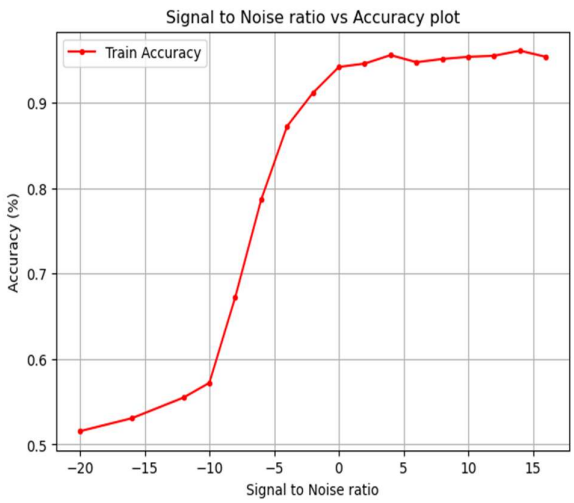
Fig. 5.2: Confusion matrix of different activation functions at SNR 16dB

Confusion Matrix Calculations At 16 dB

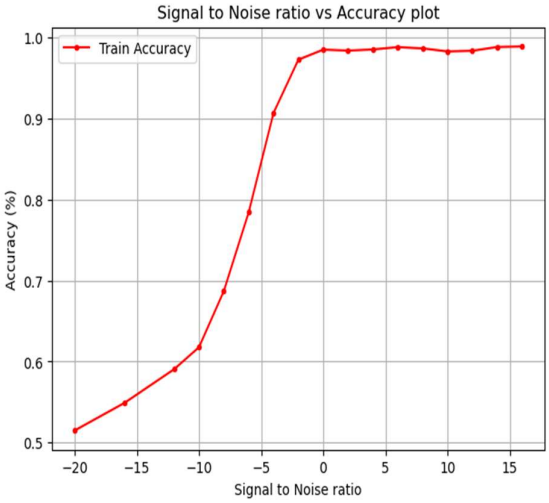
PARAMETERS	SELU	RELU	GELU	SWISH
True Negative	1040	982	1051	973
False Positive	6	1	5	8
False Negative	13	21	91	39
True Positive	1002	986	918	990
Prediction positive	1008	987	923	998
Prediction Negative	1053	1003	1142	1012
Total Population	2061	1990	2065	2010
Condition Positive	1015	1007	1009	1029
Condition Negative	1046	983	1056	981
Prediction_pos_predicted_value	0.994	0.998	0.994	0.991
False omission rate	0.012	0.020	0.079	0.038
True positive rate	0.987	0.979	0.909	0.962
False negative rate	0.012	0.020	0.090	0.037
False positive rate	0.005	0.001	0.004	0.008
True negative rate	0.994	0.998	0.995	0.991
Positive likelihood ratio	172.1	962.5	192.1	117.9
Negative likelihood ratio	77.62	47.90	11.03	26.16
False discovery rate	0.005	0.001	0.005	0.008
Negative predicted value	0.987	0.979	0.920	0.961
Diagnostic odds ratio	2.216	20.09	17.41	4.508
Accuracy	0.990	0.988	0.953	0.976

Table 1: Confusion Matrix Calculations At 16 dB

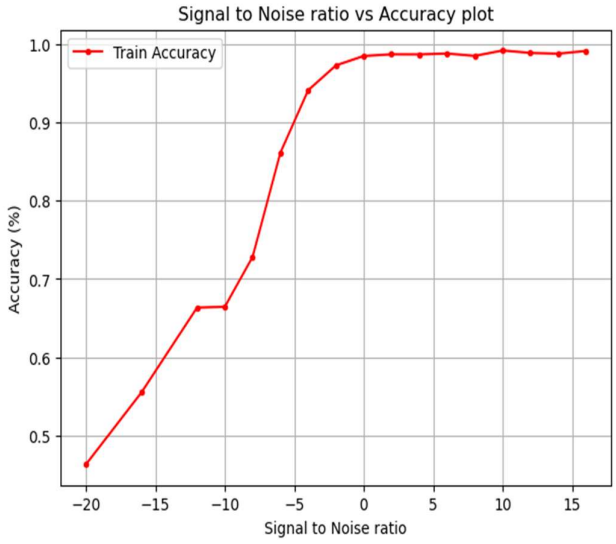
Classification accuracy vs SNR



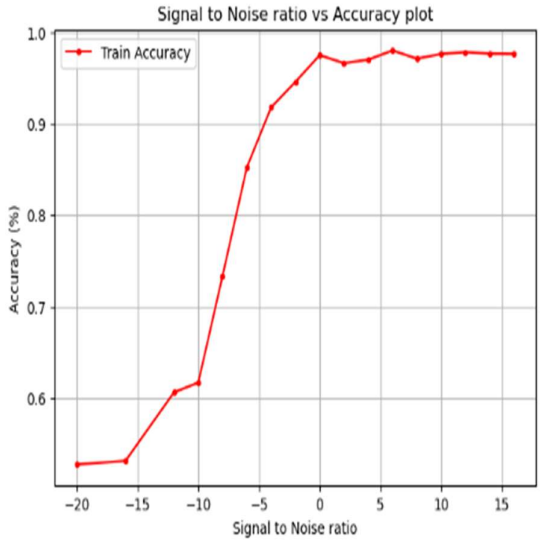
GELU



RELU



SELU



Swish

Fig. 5.3: Classification accuracy vs SNR

We are able to train multiple models with the same development but different activation functions and assess how they perform on a test set to examine the accuracy of classification of neural network models using various activation parameters. Here are some broad conclusions regarding the way Swish, ReLU, SELU, and GELU activation functions operate:

Percentage of Classification Accuracy of different values of SNR

SNR(dB)	Accuracy			
	Swish	ReLU	SELU	GELU
-20	52.66	51.49	46.35	51.55
-16	53.04	54.89	55.53	53.06
-12	60.56	59.05	66.34	55.48
-10	61.63	61.76	66.45	57.21
-8	73.33	68.7	72.81	67.24
-6	85.22	78.42	86.10	78.63
-4	91.80	90.69	94.06	87.22
-2	94.59	97.26	97.23	91.12
0	97.52	98.50	98.43	94.16
2	96.64	98.38	98.64	94.56
4	97.03	98.52	98.63	95.55
6	98.04	98.81	98.74	94.72
8	97.14	98.65	98.44	95.09
10	97.65	98.28	99.13	95.35
12	97.86	98.36	98.82	95.47
14	97.69	98.82	98.72	96.07
16	97.66	98.89	99.07	95.35

TABLE 2: The Classification Precision Of Different Activation Functions At Various Snr Values

The above table 2 represent the accuracy of classification at different SNR levels of different activation functions. From the above tables it clearly represent that at higher values of SNR, the accuracy of classification is good and at lower values of SNR, the classification accuracy is low. From the table 1 it is observed that at SNR value of -16 dB the percentage of accuracy is very low i.e., only 54.89% for ReLU. For GELU at same value of SNR the percentage of accuracy is 53.06% and for Swish at SNR value of -16 dB the accuracy is 53.04 % and finally for SELU at same value

of SNR the percentage of accuracy is 55.53 % which is higher when compared to the above values of Swish, ReLU and GELU. So we can clearly conclude that the SELU provides higher percentage of accuracy when compared to Swish, ReLU and GELU.

Accuracy Comparison Plot at 16 dB

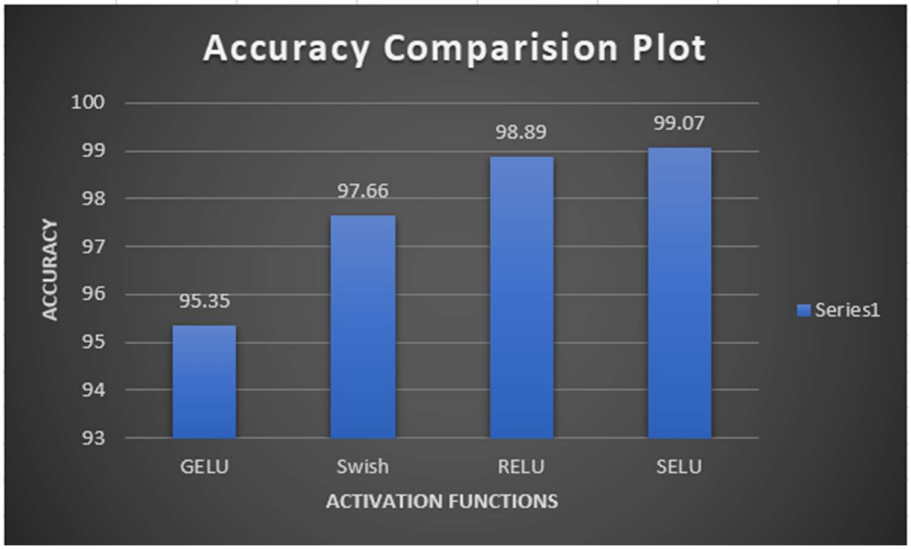


Fig. 5.4 : Accuracy Comparison Plot at 16 dB

The accuracy comparison plot at 16dB depicts the performance of different activation functions including SELU, GELU, ReLU, and Swish, in a noisy environment. The plot shows the accuracy achieved by each activation function at a signal-to-noise ratio (SNR) of 16dB, which is commonly used to represent moderate noise levels.

Among the tested activation functions, SELU (Scaled Exponential Linear Unit) stands out with the highest accuracy at 16dB SNR. SELU is known for its self-normalizing property, which helps in stabilizing the training process and reducing the impact of vanishing or exploding gradients, especially in deep neural networks. The plot shows that SELU outperforms other activation functions in terms of accuracy, indicating its effectiveness in handling noise and maintaining good performance in noisy conditions.

A 99% accuracy typically indicates that the model's predictions are correct for 99% of the samples in the dataset, without any noise or errors introduced during the evaluation process. This implies that the signal is present without any noise, and the model is able to make accurate predictions without any misclassifications.

5.2 Advantages

Deep learning, a subset of machine learning, has emerged as a promising approach for spectrum sensing due to its ability to automatically learn and extract features from raw data. Here are some advantages of using deep learning for spectrum sensing in wireless communication:

Improved Accuracy: Deep learning models can achieve high accuracy in spectrum sensing tasks by leveraging their ability to learn complex patterns and representations from large amounts of data.

Robustness to Variability: Wireless communication environments can be highly dynamic and subject to various types of interference, fading, and noise. Deep learning models can learn to adapt to such variability and perform well in different conditions, including scenarios with changing signal characteristics or varying levels of noise.

Flexibility and Adaptability: Deep learning models can be trained to work with a wide range of RF signals and modulation schemes, making them highly adaptable to different wireless communication systems and technologies.

Scalability: Deep learning models can scale effectively to handle large datasets, making them well-suited for spectrum sensing tasks that require processing of large amounts of RF data. With advancements in hardware and parallel computing, deep learning models can be efficiently implemented on high-performance computing platforms, enabling real-time spectrum sensing in practical wireless communication systems.

5.3 Applications

- Cognitive Radio
- Dynamic Spectrum Access
- Spectrum Monitoring
- Wireless Security
- Dynamic Frequency Selection
- Spectrum Prediction

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

Spectrum management issues have surfaced recently because of the expansion in wireless network users versus the availability of spectrum. In this article we used machine learning models to assign the spectrum to users in an efficient manner, which necessitates a strong and reliable means of declaring the spectrum to be unused. In this case, TP, TN, FP and FN metrics used to determine whether or not a channel was free. The positive class represents the target condition while the negative class represents the nonoccurrence of that condition. When the channel strength is positive, it means the channel's licensed user is actively engaged. If it is negative then it indicates that the channel was not utilized by licensed user. So allocation of channel can be done to unlicensed users if channel is not used.

This project explores the concept of spectrum sensing using CNN. This suggested approach shows good performance in modulation method classification at various SNR levels. This avoids the difficulty of manual feature selection at receiver side. The classification precision of a neural network can be greatly affected by the selecting of activation function. Among the four activation functions namely SELU, GELU, RELU and Swish using adadelta optimizer SELU gives highest accuracy of 99.07%.

FUTURE SCOPE:

In future this spectrum sensing method can be developed by using other supervised deep learning techniques like LSTM, GRU, and RNN and so on for identifying channel vacancy.

REFERENCES

[1] Chen, Zhibo, et al. “Deep STFT-CNN for spectrum sensing in cognitive radio” in IEEE Communications Letters 25.3, 2020.

[2] Deyu Zhang, Zhigang Chen, Haibo Zhou, “Energy Harvesting-Aided Spectrum Sensing and Data Transmission in Heterogeneous Cognitive Radio Sensor Network”, IEEE Publications, 2015.

[3] C. Liu, J. Wang, X. Liu, and Y. C. Liang, “Deep CM-CNN for spectrum sensing in cognitive radio”, IEEE Journal on selected Areas in Communications, 2019.

[4] Manse Subhedhar and Gajanan Birajdur, “Spectrum Sensing Techniques In Cognitive Radio Networks: A Survey”, International Journal of Next-Generation Networks, 2011.

[5] Captain, Kamal M., and Manjunath V. Joshi. “Spectrum Sensing for Cognitive Radio: Fundamentals and Applications” in CRC Press, 2022.

[6] Liu, Chang, et al. “Deep CM-CNN for spectrum sensing in cognitive radio” in IEEE Journal on Selected Areas in Communications 37.10 (2019).

[7] Lee, Woongsup, Minho Kim, and Dong-Ho Cho. “Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks” in IEEE Transactions on Vehicular Technology 68.3, 2019.

[8] Zheng, Shilian, et al. “Spectrum sensing based on deep learning classification for cognitive radios” in China communications 17.2. 2020.

[9] A. Mehrabian, M. Sabbaghian and H. Yanikomeroglu, “CNN-based for spectrum sensing with General Noise Models”, in IEEE Transactions on Wireless Communications, 2022.

[10] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications”, IEEE Communications Surveys Tutorials, 2009.

[11] X. Zhu, T. Wanf, Y. Bao, F. Hu and S. Li, “Signal detection in generalized Gaussian distribution noise with nakagami fading channel”, IEEE Access, 2019.

[12] E. Axel and E. G. Larsson, “Optimal and sub-optimal spectrum sensing of OFDM signals in known and unknown noise variance”, IEEE Journal on Selected Areas in Communications, 2011.

[13] B. Soni, D. K. Patel, and M. Lpez-Bentez, “Long short-term memory based spectrum sensing scheme for cognitive radio using primary activity statistics”, IEEE Access, 2020.

PROJECT WORK MAAPPING WITH PROGRAMME SPECIFIC OUTCOMES(PSOs) AND PROGRAMME OUTCOMES(POs)

Classification of Project	Application	Product	Research	Review
			✓	

Project Outcomes

Outcome: In this project Reversible image hiding using histogram shifting has been simulated by applying the conceptual knowledge of Electronics and Communication Engineering by taking the legal, ethical and economical considerations.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

The ECE Graduates will be able to

PSO1: Designing electronics and communication systems in the domains of VLSI, embedded systems, signal processing and RF communications, and applying modern tools.

PSO2 : Applying the contextual knowledge of Electronics and Communication Engineering to design, develop, analyze and test systems containing hardware and software components taking into societal, environmental, health, safety, legal, cultural, ethical and economical considerations.

PROGRAMME OUTCOMES(Pos)

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental

considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering with an understanding of the limitations activities.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project Management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-Long Learning : Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological charge.

Project Outcomes	**PROGRAMME OUTCOMES(Pos)**											**PSOs**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
Outcome	1	1	1	2	1	1	1	2	3	2	2	2	2

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 31st Mar 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 03

10.48047/IJIEMR/V12/ISSUE 03/70

Title **WIRELESS SPECTRUM DETECTION USING DEEP LEARNING FOR WIRELESS COMMUNICATION**

Volume 12, ISSUE 03, Pages: 503-509

Paper Authors
E. Vargil Vijay, K. Meghana, M. Sivaiah, K. Charan, K. Sumanth

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Wireless Spectrum Detection using Deep learning for Wireless Communication

E. Vargil Vijay[1],** Assistant Professor, SR Gudlavalleru Engineering College, Gudlavalleru
K. Meghana[2], Student, SR Gudlavalleru Engineering College, Gudlavalleru
M. Sivaiah[3], Student, SR Gudlavalleru Engineering College, Gudlavalleru
K. Charan[4], Student, SR Gudlavalleru Engineering College, Gudlavalleru
K. Sumanth[5], Student, SR Gudlavalleru Engineering College, Gudlavalleru
[1] **Corresponding author E-mail: **Vargilvijay@gmail.com**

Abstract:
The main focus of this article is Spectrum sensing for efficient data transfer in wireless communications. Currently, there is a need for an effective wireless communication system because the serviceable spectrum is not sufficient due to exponential growth in the number of wireless device users. One approach for effectively using the spectrum is through Radio frequency sensing. Spectrum-aware radio relies on Radio frequency sensing to detect available spectrum for better usage and to reduce harmful interference with licensed users. For spectrum sensing, there are currently established techniques based on energy detection, but the development of machine learning has made spectrum sensing more efficient. A machine learning-based model can perform better than traditional methods. To identify the existence of a licensed user, we suggest a machine learning spectrum sensing technique employing a Convolutional Neural Network model in a Gaussian environment. Such that if the spectrum is not being utilized by the licensed user then the secondary user can benefit from it. This methodology will be very helpful in resolving the spectrum scarcity problem in wireless communications.

Key Words: *Spectrum sensing, Spectrum utilization, Energy detection, CNN.*

INTRODUCTION:

Wireless networks are now widely used and considered essential in today's life due to the rapid development growth of its user base and range of applications [1]. The increasing demand for radio spectrum is beyond the capabilities of the present spectral management model is to keep up. To reduce this shortage, a variety of spectrum reuse solutions have been proposed. By enabling unlicensed users opportunistic access to the spectrum, it is possible to address the issue of spectrum utilization with the help of Spectrum-aware radio [2].

According to the research that has been done, Cognitive Radio has many more benefits than only better spectrum management. When it comes to the spectrum shortage, Cognitive Radio was the premier solution to address the problem of spectrum scarcity because of its many benefits, including enhanced throughput and novel services in radio equipment and higher spectral efficiency [3].

Cognitive radio technology is an interesting strategy for solving the problem of underutilized wireless spectrum. Cognitive radio can adapt the settings of its transmitter to the conditions in which it is operating [4]. Cognitive radios are developed to promote efficient use of the radio spectrum and to ensure that all users of the network have access to highly reliable communication capabilities at all times. Spectral sensing, Spectral management, Spectral sharing, and Spectral mobility are the four cornerstones of cognitive radio's architecture. The purpose of spectrum sensing is to identify

authorized users and obtain information about available frequencies [5]. Spectrum sharing is the process of allocating scarce radio frequencies among competing users in a cost-effective manner. The goal of spectrum mobility is to keep communications running smoothly even when they are upgraded to a higher quality frequency range [3].

Literature Survey

Among all the studies, convolutional neural networks (CNNs) have emerged as the clear frontrunner. Studies by Captain [5] et al. discussed the most popular Radio frequency sensing methods which are energy disclosure and matched disclosure and they found that Receiver Operating Characteristic curves are being created by representing either detection chance versus false alarm probability. Given a bell-shaped distribution with zero mean and one standard deviation. Studies by Lee [7] et al. discussed about cooperative Radio frequency sensing in a reinforcement learning method and they found that Conventional CSS techniques perform worse than DCS, especially in challenging sensing environments. Further the run time of DCS is larger than those of conventional methods. It can be improved by decreasing the computation time.

SPECTRUM SENSING:

Radio signal detection in a specific frequency range is known as spectrum sensing. In dynamic spectrum access systems, where wireless devices can opportunistically access unused frequency bands, this technique is extremely important for facilitating optimal use of spectrum resources [1]. Spectrum sensing can be used in the context of data transfer to identify channels with sufficient bandwidth. Primary transmitter detection is used to describe the process of identifying the existence or vacancy of a primary transmitter signal in a particular frequency band [6].

Spectrum sensing can be accomplished by a variety of designs, namely cyclostationary feature identification, and compressive sensing. Different methods have different complexity levels, precision levels, and computational needs [3]. Particularly in cases where spectrum resources are scarce or otherwise congested, spectrum sensing plays a crucial role in ensuring efficient and reliable data transmission via wireless networks. Since models based on machine learning to outperform more traditional approaches, that are increasingly being used to improve results. To identify licensed user, we come up with a Convolutional Neural Network model operating in a Gaussian environment for spectrum sensing [4].

PROPOSED MODEL

As a subset of Deep Learning architecture Convolutional Neural Networks (CNNs) are frequently employed in image classification and recognition applications. It has several layers such as filter layers, down sampling layers and dense layers. The convolutional layer employs kernels on the input feature in order to extract features. During training, a Conv layer employs a set of learnable filters to the input feature and produces a set of feature maps.

1. Input Layer: Here, we feed information into our model at the input layer. Our data contains a total of n characteristics, therefore n neurons were present in this layer.
2. Hidden Layer: It consists of multiple convolutional and pooling layers. The convolution layers apply a set of learnable filters to the input feature, and build a set of feature maps. The pooling layer diminishes the structural dimensions of the feature maps and the output of the last hidden layer is unrolled and given to fully connected layers and the turnout of this layer is fed into sigmoid layer.
3. Output Layer: To determine the probability score for each class, a logistic function like sigmoid is applied to the output of the latent layer.

Next stage is, feed forward, involves supplying the model with data in order to

obtain the layer-by-layer output. Then, we use an error function to calculate the error. Then, we back produce the model after computing variables. In general Back propagation is the process that is used to reduce waste. In this article swish activation function with lecun uniform and lecun normal kernel initializers have been proposed, and in the final stage for classification purpose sigmoid activation has been used with adadelta optimizer.

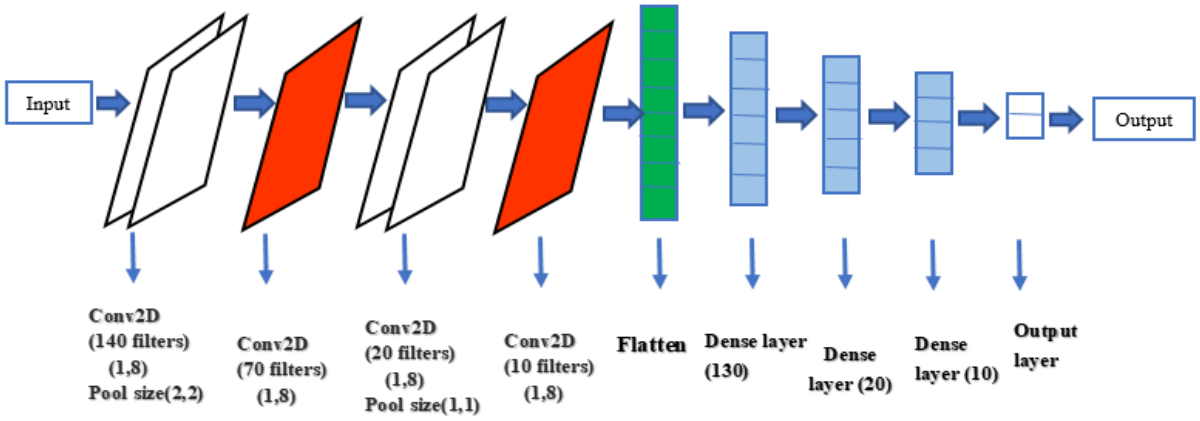


Fig.1Proposed diagram based on CNN

The above diagram illustrates the working of the proposed model. The data set is the input to the convolution layer, which typically represents the pixel matrix of the data. The input layer is fed to 2-dimensional convolutional layer with 140 filters, each of which has size of 1x8 (meaning of filter size is 1 along the vertical dimension and 8 along the horizontal dimension). The pool size (2,2) indicates that after the convolutional operation, a max-pooling operation with a pool size of 2x2 is applied to the output layer of the convolutional layer. Overall, this layer is meant to glean 140 sets of feature sets from the input data, each of which corresponds to a different filter, and then reduce the spatial size of the feature maps through max-pooling. The next layer is the 2 dimensional convolutional layer with 70 filters each of which has size 1x8. This layer is designed to extract 70 sets of features from the input data each of which corresponds to a different filter. The spatial size of the output feature maps from the convolutional layer is already small enough. So there is no need to apply max-pooling to further reduce the size. The next layer is Conv2D which consists of 20 kernels which has size 1x8. The pool size of 1x1 is fed to decision layer of feature extractor. Actually this type of pooling operation doesn't actually perform any pooling or down sampling of the input data. Instead, it simply passes the input data through unchanged. This is fed to Conv2D with 10 kernels which has size 1x8 and there is no max-pooling layer because the size is already reduced. The layer next to the 2 dimensional Convolutional layer is the flatten layer. A conv2d layer is used to extract features and generate 2D feature maps as output. In order to pass this information to fully connected layer that expects 1D input, the feature maps need to be flattened into a vector. It is fed to a dense layer with 130 neurons, it means that the flattened vector is being treated as a 1D input to a fully connected layer with 130 neurons. The flattened vector contains the feature information that has been extracted by the previous layers in the network. Each neuron in the dense layer takes thus input vector and performs matrix multiplication with a weight matrix followed by a bias term, to produce an output value. The weights and biases in the layer are learned through backpropogation

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

during training. It is fed to a dense layer with 20 neurons and then fed to a dense layer with 10 neurons. The purpose of adding another dense layer in this way is to further process the extracted feature information from the previous layers and to provide more opportunities for the network to learn high-level representations of the input data. It improve its ability to make accurate predictions on new data. Finally the dense layer with 10 neurons is fed to the output layer which consists of one or more neurons depending on the specific task. For multi-class classification the output layer may consist of multiple neurons, each representing a different class, with a sigmoid activation function to produce a probability distribution over the 2 classes. For developing this architecture model in this article deepsig dataset that is radioml 2016.04.C has been considered and was downloaded from the following RF datasets for Machine Learning website: https://www.deepsig.ai/datasets. Overall the result of the network's computation on the input data and can be used for a variety of tasks, including prediction, classification, and regression. The output of the network may be scalar or vector representing the predicted numerical value or values.

RESULTS

Here the performance of a classification model on a dataset with binary labels can be summarized using various metrics, including TP, TN, FP, FN, accuracy, precision, recall. These metrics are typically presented in a Table 1.

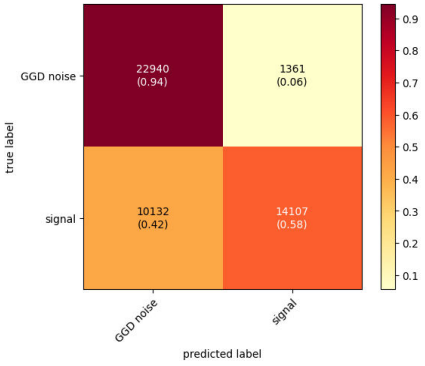
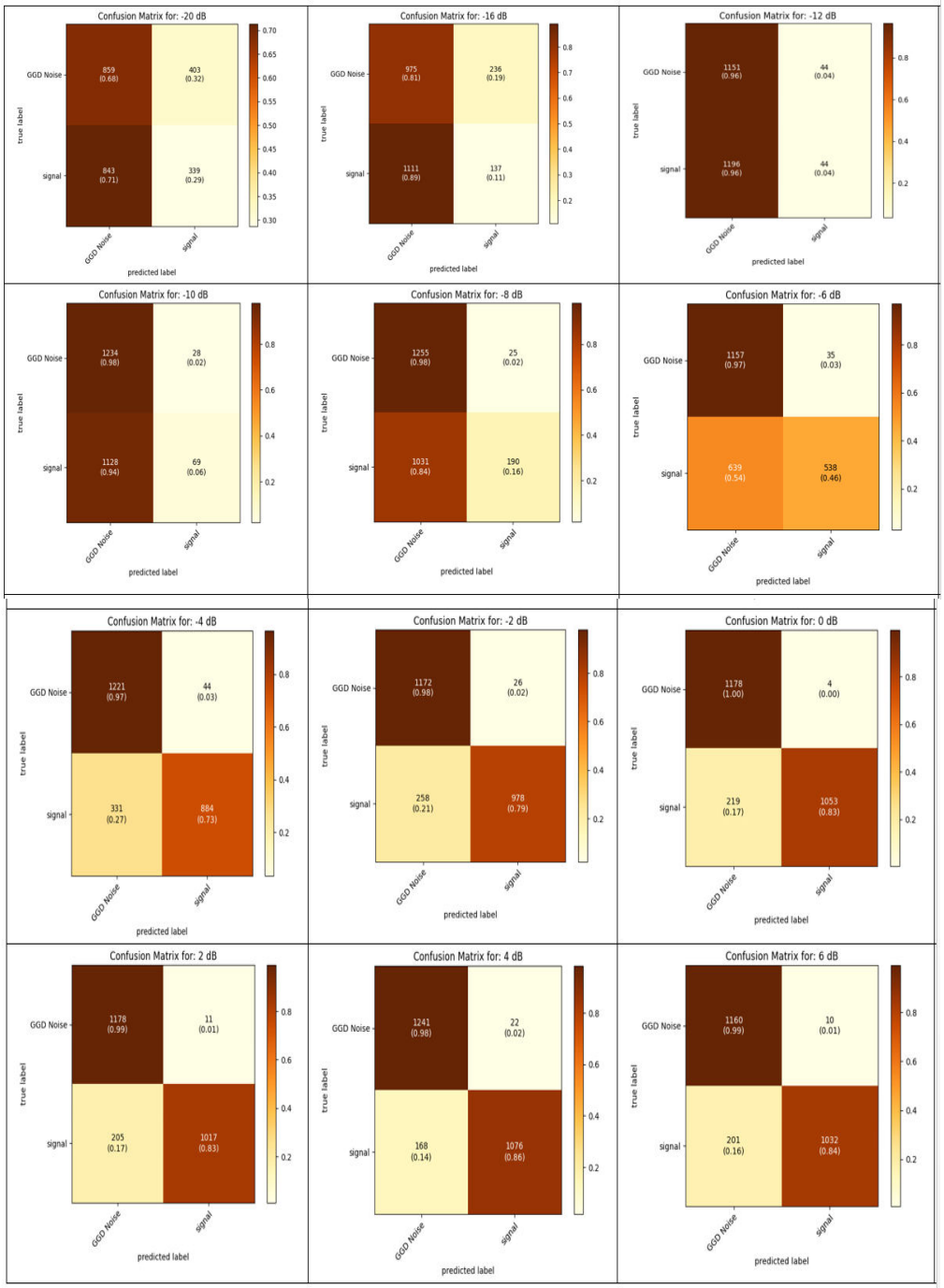


Fig. 2 Confusion Matrix

		True Condition			
	Total Population	Condition Positive	Condition Negative	Prevalence	
Predicted Condition	Prediction Positive	True Pos (TP) 1361	False Pos (FP) (type 1 error) 1361	Precision Pos Predictive value 0.912	False Discovery Rate 0.087
	Prediction Negative	False Neg (FN) (type 2 error) 10132	True Neg (TN) 22940	False Omission Rate 0.306	Negative Predictive Value 0.69
	Accuracy 0.763	Sensitivity (SN), Recall, Total Pos Rate 0.58	Fall-Out, False Positive Rate FPR 0.056	Pos Likelihood Ratio LR+ 10.35	Diagnostic Odds Ratio DOR 4.6
		Miss Rate, False Neg Rate FNR 0.418	Specificity (SPC), True Neg Rate 0.943	Neg Likelihood ratio LR- 2.25	

Table.1 Calculations for Confusion Matrix

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org



International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

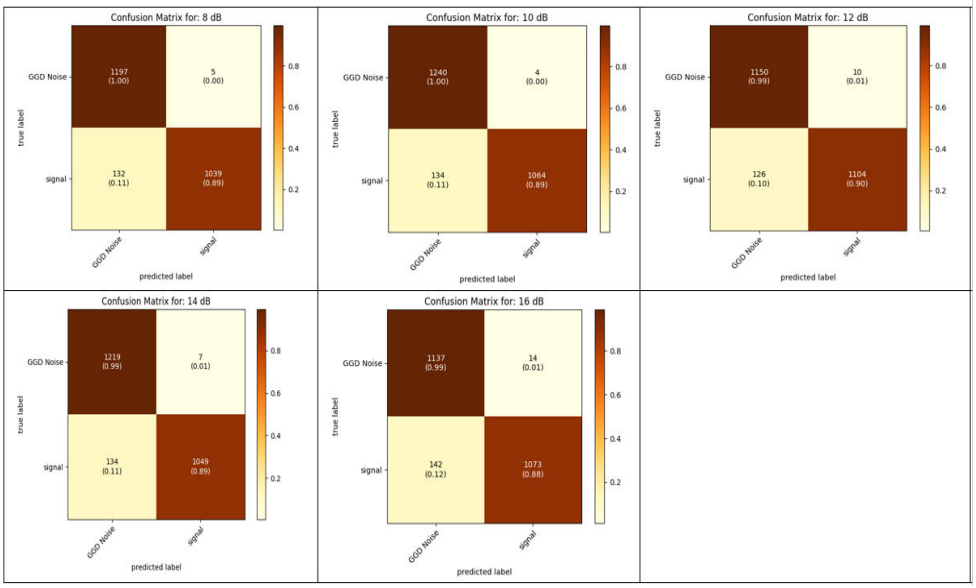


Fig. 3 Confusion matrix values for -20 dB to 16 dB

We summarize the performance of a model on classification task of detecting a signal in the presence of noise based on Table 1. The positive class corresponds to positive class and negative class corresponds to the absence of signal. Here the model has an accuracy of 0.76, indicating that it correctly that it correctly predicts the class of 76% of the samples in the test set represents in Table 1.

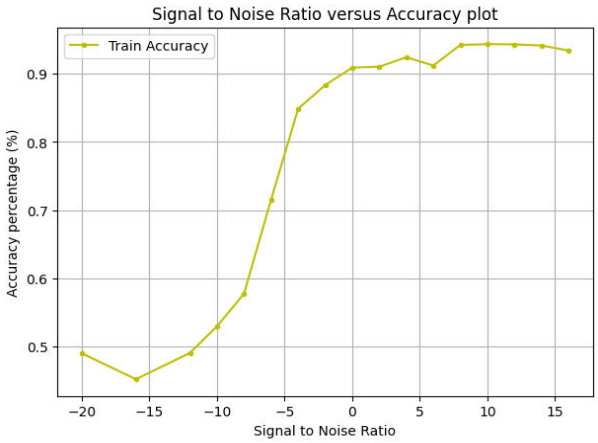


Fig. 4 Accuracy vs SNR

CONCLUSION

Spectrum management issues have surfaced recently because of the expansion in wireless network users versus the availability of spectrum. In this article we used machine learning models to assign the spectrum to users in an efficient manner, which necessitates a strong and reliable means of declaring the spectrum to be unused. In this case, TP, TN, FP and FN metrics used to determine whether or not a channel was free. The positive class represents the target condition while the negative class represents the nonoccurrence of that condition. When the channel strength is positive, it means the channel's licensed user is actively engaged. If it is negative then it indicates that the channel was not utilized by licensed user. So allocation of channel can be done to unlicensed users if channel is not used.

REFERENCES

[1] Chen, Zhibo, et al. "Deep STFT-CNN for spectrum sensing in cognitive radio" in IEEE Communications Letters 25.3, 2020.
[2] Deyu Zhang, Zhigang Chen, Haibo Zhou, "Energy Harvesting-Aided Spectrum Sensing and Data Transmission in Heterogeneous Cognitive Radio Sensor Network", IEEE Publications, 2015.
[3] C. Liu, J. Wang, X. Liu, and Y. C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio", IEEE Journal on selected Areas in Communications, 2019.
[4] Manse Subhedhar and Gajanan Birajdur, "Spectrum Sensing Techniques In Cognitive Radio Networks: A Survey", International Journal of Next-Generation Networks, 2011.
[5] Captain, Kamal M., and Manjunath V. Joshi. "Spectrum Sensing for Cognitive Radio: Fundamentals and Applications" in CRC Press, 2022.
[6] Liu, Chang, et al. "Deep CM-CNN for spectrum sensing in cognitive radio" in IEEE Journal on Selected Areas in Communications 37.10 (2019).
[7] Lee, Woongsup, Minho Kim, and Dong-Ho Cho. "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks" in IEEE Transactions on Vehicular Technology 68.3, 2019.
[8] Zheng, Shilian, et al. "Spectrum sensing based on deep learning classification for cognitive radios" in China communications 17.2. 2020.
[9] A. Mehrabian, M. Sabbaghian and H. Yanikomeroglu, "CNN-based for spectrum sensing with General Noise Models", in IEEE Transactions on Wireless Communications, 2022.
[10] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications", IEEE Communications Surveys Tutorials, 2009.
[11] X. Zhu, T. Wanf, Y. Bao, F. Hu and S. Li, "Signal detection in generalized Gaussian distribution noise with nakagami fading channel", IEEE Access, 2019.
[12] E. Axel and E. G. Larsson, "Optimal and sub-optimal spectrum sensing of OFDM signals in known and unknown noise variance", IEEE Journal on Selected Areas in Communications, 2011.
[13] B. Soni, D. K. Patel, and M. Lpez-Bentez, "Long short-term memory based spectrum sensing scheme for cognitive radio using primary activity statistics", IEEE Access, 2020.
[14] M. Bkassiny, Y. Li and S. K. Jayaweera, "A Survey on Machine-Learning Techniques in Cognitive Radios", in IEEE Communications Surveys and Tutorials, 2013.
[15] Xie, Jiandong, "Deep learning-based spectrum sensing in cognitive radio: A CNN LSTM approach", IEEE Communications Letters, 2020.