STUDIES ON SOME MACHINE LEARNING AND SIGNAL PROCESSING TECHNIQUES FOR CONDITION MONITORING APPLICATIONS

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

Submitted by

Debasmit Dey

Examination Roll Number: M4ELE22018

Registration Number: 154008 of 2020-21

Under the guidance of

Prof. Debangshu Dey

Electrical Engineering Department

Faculty of Engineering and Technology

Jadavpur University

Kolkata-700032

West Bengal, India

FACULTY OF ENGINEERING & TECHNOLOGY

JADAVPUR UNIVERSITY

Kolkata-700032

Certificate of Recommendation

This is to certify that the thesis entitled, **"STUDIES ON SOME MACHINE LEARNING AND SIGNAL PROCESSING TECHNIQUES FOR CONDITION MONITORING APPLICATIONS",** has been carried out by **Debasmit Dey (Roll no. 002010802018)** under my guidance and supervision and be accepted in partial fulfillment of the requirements for the degree of **Master of Engineering in Electrical Engineering** under the Department of Electrical Engineering, Jadavpur University, Kolkata-700032.

Dr. Debangshu Dey

Associate Professor, Department of Electrical Engineering

Jadavpur University, Kolkata-700 032

Prof. Chandan Mazumdar

Dean

Faculty of Engineering and Technology

Jadavpur University,Kolkata–700032

Prof. Saswati Mazumdar

Head of the Department

Department of Electrical Engineering

Jadavpur University, Kolkata-700 032

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Kolkata-700032

Certificate of Approval

The foregoing thesis, entitled **"STUDIES ON SOME MACHINE LEARNING AND SIGNAL PROCESSING TECHNIQUES FOR CONDITION MONITORING APPLICATIONS"** is hereby approved as a creditable study of an engineering subject conducted and presented satisfactorily to warrant its acceptance as a precondition to the degree for which it was submitted. It is notified to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed and conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Committee on Final Examination
for Evaluation of the Thesis

_____

Signature of External Examiner

_____

Signature of Supervisor

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains a literature survey and original research work by the undersigned candidate, as part of his Master of Engineering in Electrical Engineering studies under the department of Electrical Engineering.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name: Debasmit Dey

Exam Roll Number : M4ELE22018

Thesis Title: **"Studies on some Machine Learning and signal processing techniques for condition monitoring applications"**

Signature of Candidate

ACKNOWLEDGEMENTS

This thesis entitled **"STUDIES ON SOME MACHINE LEARNING AND SIGNAL PROCESSING TECHNIQUES FOR CONDITION MONITORING APPLICATIONS"** is the result of the work whereby I have been accompanied and supported by many people, my guide, my friends, and lab seniors. It is a pleasant aspect that now I have the opportunity to express my gratitude to all of them.

First and foremost, I would like to express my sincere gratitude to my thesis supervisor **Prof. Debangshu Dey**, Associate Professor of Department of Electrical Engineering, Jadavpur University for his valuable guidance, insightful suggestion and support while conducting this thesis work as well as in the writing of this thesis. I have been very fortunate to have a guide like him. His positivity, confidence, and ideas help me to complete my thesis and he guides me as a guardian.

I am thankful to my fellow project mates, friends, technical and non-technical staffs of Jadavpur University who have helped me during the tenure of my thesis work.

I want to express my gratitude to my parents and family also, as, without their sacrifices, I can't do anything. And their invaluable love, encouragement, and support make me, whatever I am today.

Debasmit Dey

Jadavpur University

Kolkata-700032, West Bengal

TABLE OF CONTENTS

3. **BEARING FAULT CLASSIFICATION USING DEEP LEARNING TECHNIQUES**

List of Figures

List of Tables

ABBREVIATION

PD	Parkinson's Disease
HC	Healthy Controls
ML	Machine Learning
DL	Deep Learning
EEG	Electroencephalography
SVD	Saarbrucken Voice Database
WT	Wavelet Transform
DWT	Discrete Wavelet Transform
XWT	Cross-wavelet Transform
MFCC	Mel-frequency cepstral coefficients
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
BN	Batch Normalization
SVM	Support Vector Machine
KNN	k-Nearest Neighbors
AUC	Area under the curve
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
LSTM	Long-short term memory
GRU	Gated Recurrent Unit

CHAPTER 1

INTRODUCTION

In recent times, Deep Learning and Machine Learning techniques along with signal processing have extensively used in various applications such as early detection of various diseases, satellite image segmentation, self-driving cars, natural language processing, object detection, fault diagnosis of motor, medical image classification, weather forecasting etc. Among these, biomedical field and condition monitoring of various machines in industry are two major areas for the application of Artificial Intelligence. Artificial intelligence and machine learning have been used effectively in detection and treatment of several dangerous diseases, helping in early diagnosis, and thus increasing the patient's chance of survival. Deep learning has been designed to analyze the most important features affecting detection and treatment of serious diseases. Also various machine learning and deep learning algorithms have been used in industry for fault diagnosis, predictive maintenance of machines to increase the operational reliability and reduce costs. In this thesis three condition monitoring applications have been taken within which one application was from the biomedical field and other two were based on fault diagnosis.

The structure of the thesis is as follows:

(i) Chapter 2 has been proposed the Parkinson's disease classification problem using audio signals. Several Deep Learning algorithms (VGG16, VGG19, DenseNet-121 and Xception) have been introduced in this chapter to extract the features from Wavelet and Cross-wavelet scalograms which represented the time-frequency information about the audio signals. Multiple Machine Learning algorithms have been used for the classification from the extracted features. A majority voting ensemble technique has also been introduced in this chapter.

(ii) Chapter 3 demonstrated bearing fault classification problem using Deep Learning techniques namely Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Long-short term memory (LSTM), Gated Recurrent Unit (GRU) and ConvLSTM. For this purpose, various time-domain and statistical features have been extracted which were fed into these deep learning models. A Weighted Average Ensemble technique has been deployed using these models to improve the model performance.

(iii) Chapter 4 represented the bearing fault predictive maintenance problem which was stated that after how many cycles machine was going to fail. For this purpose, RMS value has been taken as condition indicator of the machine which has provided the information about machine's health. LSTM has been used for this prediction problem because they are very good at handling time-series sequence. KernelRidge regression has also introduced for comparing its result with LSTM based model.

(iv) The thesis has been concluded in chapter 5.

CHAPTER 2

Parkinson's disease classification with Wavelet, Cross-wavelet Transform and Transfer learning using Machine Learning models

2.1 INTRODUCTION

Parkinson's disease (PD) is a slow and advanced neurodegenerative brain disease that damages brain cells. The disease is referred to as progressive since it gets worse with time and neurons in the brain are degenerated during this disease. PD impacts over 10 million patients worldwide. Dopamine neurons are a specific type of neuron which are lost during PD. Dopamine regulates the control and flow of body movements. Dopamine, a tiny signaling substance produced by dopamine neurons, is crucial for enabling people to move normally. The degeneration of dopamine-producing cells results in the decreased level of dopamine production [1]. This causes the symptoms of PD. As PD progresses people lose control over their movement and coordination [2] . Although PD may be diagnosed at younger ages, the disease usually affects the age group of 40 to 75 years and most commonly the people over the age of 60 years. People with 55-75 years of age are more vulnerable to PD. Still, there has been no treatment which can prevent the disease's progression [3].

There are four main symptoms of PD. The first symptom is tremor. Tremor usually starts in hands or arms; it may appear at rest and disappear when the affected body part is moved, yet it may be persistent in certain postures. The second symptom is slowness of movement, which is called bradykinesia. Slowness in movement appears in movements of arm, leg and body, reduces the ability to swing arms while walking, causes walking problems and leads to decrease in facial expression due to deformed

facial mimic muscles. The third symptom is stiffness in arms, legs and trunk, or in other words, rigidity. When a person bends their body parts, such as their arm, leg, or wrist, they may feel stiffness. The fourth symptom is postural instability, which causes increased forward bend of torso and shortened step length. Balanced problems are commonly observed in this symptom. These are four cardinal clinical features. Apart from these four cardinal clinical features, other symptoms may also appear. For example, the patient may have abnormal, smaller or shaky handwriting or may experience problems with writing signature. The patient may also suffer from slurred or slow speech. The main symptoms may be accompanied by reduced sense of smell, constipation, and changes in blood pressure. Another symptom is Rapid eye movement (REM) sleep behavior disorder which is usually defined as the dream enacting such as talking, yelling, kicking or punching during sleep. Shakiness is also a symptom of Parkinson's disease.

The Hoehn and Yahr (H&Y) scaling and the Unified Parkinson's disease rating scale (UPDRS) and are two most common assessment scale for measuring and tracking the progression of PD [4]. H&Y scaling is categorized the progression of PD over time into five different stage according to the severity of the disease. Although progress is often gradual, it might differ from patient to patient. In first stage mild symptoms are appeared including tremor, other motor symptoms namely shakiness, stiffness, slowed down movements and postural instability which produces disability at one side of the body. Stage 2 symptoms make daily tasks more challenging, and there are movement issues on the body. Even slight movements can have an impact on posture and walking. Movements slow down a bit in stage 3. Patients are more likely to lose their balance, increasing their risk of falling. In stage 4, severe movement problems necessitate the need for a walker. Patients require significant assistance with daily tasks. Stage 5: The person requires a wheelchair and assisted living due to significant balance and mobility issues. Additionally, they might experience observable mental changes, delusions, or hallucinations.

During the treatment of PD, numerous factors such as the patient's age, stage of the disease and adverse effects of the medications used are taken into consideration. Early

intervention is critically important; the treatment should be started before dopamine reserves are depleted. There are certain medications are used to elevate and maintain dopamine levels. The progress of disease may be accompanied by findings such as depression and dementia. Such findings also need to be treated by medication. At this moment, there is no proper cure or treatment available for Parkinson’s disease. Treatment can only be done during the beginning part of the disease. The diagnosis cost is very high and finding early signs of Parkinson's disease is quite challenging because early symptoms are very similar with the other aging problems. Therefore various Deep Learning (DL) and Machine Learning models are used for the early detection of PD.

PD is also affected the alteration of voice and speech, 90% of PD patients face this kind of speech disability and vocal disorder in their earlier stages of PD [5]. PD can be affected speech in many forms such as pronounciation, spoken language output, production of voice etc [6–9]. Vocal chord atrophy and abnormalities in Parkinson's-related hypokinetic dysarthria have been found in a number of unique patterns that can be seen with direct laryngoscopy. [10]. The most typical signs of Parkinson's disease include tremor in the voice, hoarseness, feeble and repetitive speech, air shortage, incorrect articulation, and silent voice. Slow initiation can also cause latency in response, which may be followed by rushes of speech. Over the course of the illness, a decline in reading rate is frequently noticed [11,12]. Voice analysis and determination of some speech parameters namely changes in voice frequencies (jitter), vocal cord pressure at the opening, amplitude of the volume, amplitude difference between voice cycles (shimmer), and so on, can be used to study speech and voice. PD patients had a shorter phonation time, more jitter, a smaller pitch range with higher threshold pressure during phonation [13]. Detection of PD using Audio based classification is a very popular technique in recent researches. Audio features were extracted by using various audio signal processing techniques like Mel Frequency Cepstral Coefficients (MFCC) from the audio signals of PD patients and then fed those extracted features to different machine learning algorithms for detection of PD.

In this chapter, Wavelet Transform (WT) and Cross-wavelet Transform (XWT) have been applied on the audio signals for time-frequency representation. WT and XWT provide the time-frequency scalogram images. Then pre-trained models like VGG16, VGG19, DenseNet-121 and Xception architecture have been applied individually on these scalogram images for the features extraction. The extracted features are then fed into various ML algorithms like Support Vector Machine (SVM), Logistic Regression, k-nearest neighbors (KNN) and Random Forest. The audio data set which contains the audio recordings of 21 healthy controls (HC) and 18 patients with Parkinson's disease (PD) is utilized as a training set to examine how well these four machine learning methods perform in terms of important metrics including accuracy, recall, precision, f1-score and area under the curve (AUC score). The outcomes are highly competitive, and they can be applied to both detection and therapy. Further, an ensemble model (majority voting ensemble) using these four machine learning models is developed in this chapter which demonstrates higher performance compared to individual machine learning models. The detection models presented in this chapter can further be employed for other different diseases. Eight feature extraction based techniques have been proposed in this chapter, namely WT+VGG16, WT+VGG19, WT+DenseNet, WT+Xception, XWT+VGG16, XWT+VGG19, XWT+DenseNet, XWT+Xception.

The remaining of the chapter is laid out as follows. A description of the literature review is presented in section 2.2. The description of the collected audio data has been mentioned in section 2.3. In section 2.4, a brief description of the proposed methodology has been mentioned. This section also contains various pre-processing steps on the audio data such as signal processing and features extraction. The experimental results of different feature extraction techniques and machine learning models have been discussed in section 2.5. Section 2.6 contains conclusion and future scope of the proposed work.

2.2 LITERATURE REVIEW

For decades, scientists have been researching the impact of Parkinson's disease on the brain. EEG (Electroencephalography) recordings have been employed in many of these investigations to track the changes brought on by Parkinson's disease. Schlede discovered a relationship between cognitive deficits and the overall EEG (GTE) score in individuals with Parkinson's disease in 2011 [14]. GTE score is defined as a rating system for EEG examinations that can be used to assess various dementias.

Swann assessed the SSRT (Stop Signal Reaction Time) of 15 PD vs 15 healthy persons using EEG recordings, ERP data and time-frequency representation in the same year [15].

Klassen employed EEG recordings as a prognostic indicator for dementia progression in Parkinson's disease patients [16].

Yuvaraj et al. introduced the intensity and frequency of EEG readings during emotional to distinguish between PD and healthy controls in 2014. The researchers applied 14-channel EEG readings from 20 PD and 30 healthy persons [17]. They have achieved 95% accuracy in their result. Later that year, they released another work in which they classified the different emotions based on the EEG readings of 20 Parkinson’s disease patients and 20 healthy controls. They have classified six basic human emotions such as happy, sorrow, anger, disgust, fear, and surprise. Two ML algorithms such as SVM and KNN have been deployed to detect the disease using three HOS features. These features are bi-spectral magnitude average, normalized bi-spectral entropy and normalized bi-spectral squared entropy [18]. They have also introduced ten-fold cross-validation on testing set for increasing robustness in the result.

Yuvaraj et al. estimated 13 features from the previously used dataset to classify between PD patients and healthy controls, building on their earlier work from 2016. They reported accuracy of 99.62 percent and specificity of 99.25 percent [19].

In 2017, Liu et al. [20] employed a three-way decision model to detect the PD patients using EEG signals. They have applied the Discrete Wavelet transform (DWT) on EEG signals and extracted sample entropy which was used as a feature. They have used EEG reading from 42 PD and 42 healthy persons in their work. This work contains three stages for the analysis of EEG signals and then three-way decision model was used for the detection of PD.

DL applications in EEG signals have received some attention in recent years [21]. Oh et al. [22] employed a Convolutional Neural Network (CNN) for the PD classification in 2018. Their proposed CNN Model contained 13 layers which was used for training the dataset. Their algorithm had a specificity of 91.77 percent and an accuracy of 88.25 percent.

For HC and PD patients, Obukhov et al. computed the time–frequency characteristics of central EEG electrodes, interruption of the main rhythm and the appearance of rhythms within the frequency range of 4-6 Hz were employed as indicators for the early diagnosis of PD [23].

The difficulty to access the SN region in the brain is one of the key challenges in diagnosing PD. Many researchers have recently tried to mimic the electrical potential using actual EEG data from Parkinson's patients, at the basal ganglia [24]. Following the simulation, they classified distinct stages of the disease based on the combined strength of beta and alpha rhythms. This research has shown while participants engaged in activities that release dopamine in order to improve categorization accuracy. The classification will be more challenging if the dopamine level is low; nevertheless, rest state data is considerably easier to get. Instead of focusing on channel localisation, we make use of far more advanced characteristics.

The analysis of voice signals to obtain acoustic characteristics has been presented as a scientifically valid and painless treatment for PD identification in several publications [25]. Using machine learning methods, Sakar BE, et al. [26] discovered that prolonged vowels carry adequate Parkinson’s Disease discriminative information.

Wodzinski M [27] used Saarbrucken Voice Database (SVD) and PC-GITA database to detect PD using a transfer learning (ResNet) based architechture. For this work, each audio signal was taken as 0.5 second of duration. Audio spectrograms from the PC-GITA database [28] utilised for categorization. Then pre-trained ResNet architecture has been deployed for the classification. The network has been trained using ImageNet and SVD database. For all layers, Stochastic Gradient Descent has been used with learning rate of 0.0005 and for the dense layers, learning rate was taken as 0.008. For pre-training the network with SVD database momentum and minibatch size have taken as 0.95 and 64 respectively. The cross-entropy loss was employed as the loss function. This proposed model was pre-trained for 2000 epochs. Testing accuracy shown in this paper is more than 90%.

DL models are coupled with ML approaches in Zahid, et al. [29] to detect PD patients. They have used short-time Fourier transform to find the spectrograms from the audio recordings of HC and PD patients. Then features were extracted from the spectrograms using Alexnet architecture. These extracted features were then fed into SVM, Random Forest and multilayer perceptron to detect the PD. In this work features were also directly extracted from the audio signals, namely 12 Mel-Frequency Cepstral Coefficients (MFCCs), various time-domain features (maximum value, standard deviation, energy and average value) of phonation and prosody. The features were then fed into machine learning models. With this methodology they have achieved the maximum accuracy of 84.6% with Random Forest.

The detection of vocal disorders in individuals with PD is proposed by Trinh et al. [30]. In this work, The SVD and The Spanish Parkinson's Disease Dataset (SPDD) have been used. Short-time Fourier transform (STFT) has been introduced to get time-frequency spectrogram images. Then input spectrogram images of dimension 28x28x3 were passed through three convolution layers followed by one max-pooling layer. In their proposed model, first convolutional layer contains 16 filters with kernel size of 3x3 and 'same' padding. The remaining two convolutional layers contains 32 and 64 filters respectively with same kernel size and 'same' padding. The image size at the output of convolutional operation was 28x28x64. Then the output activation map has

been passed through one max-pooling layer with kernel size of 2x2. Output image after this max-pooling operation was flatten and total 50,176 features were extracted which were fed into three dense layers with 128, 64 and 1 neurons respectively. This CNN model which has been deployed in this work provided accuracy of 95%.

2.3 DATASET DESCRIPTION AND PREPARATION

The audio dataset which was used for PD classification was obtained from the Mobile Device Voice Recordings at King's College London (MDVR-KCL) [31]. From September 26 to September 29, 2017, this dataset has been collected at King's College London (KCL) Hospital, Denmark Hill. For the voice recordings, an examination room was used with a (10m x 10m) room space and a 500-millisecond reverberation duration. Because the voice recordings were made in a practical situation of making a phone call like the participant places the phone close to his or her favored ear and speaks into the microphone.

The audio recordings of 21 healthy controls (HC) and 18 patients with PD are present in this dataset. The audio recordings were recorded by using a Smartphone (Motorola Moto G4). To execute the recorded audios on the smartphone, a "Toggle Recording Software" was created which has the same features as the i-PROGNOSIS smartphone app's voice recording module. It is also available as an independent android app. Because of that the speech collecting service runs on the recording device's background and also activates voice recordings based on the Smartphone's on- and off-hook signals. The audio recordings in the database are stored in the WAVE (.wav) file format. The audio recordings are comprised of two different text passages, which were mentioned in description of the dataset.

In this chapter, only those audio recordings were used which contains the first text passage. Each audio recording was assessed by using H&Y scale, UPDRSII-5 and UPDRSIII-18 scale ratings. H&Y rating was categorized the PD patients into three stages (2, 3 and 4) and remaining 21 Healthy Controls (HC) is labeled as '0' in the

dataset. Audio recording files present in this dataset are represented by the patient's id, H&Y rating, UPDRSII-5 and UPDRSIII-18 rating respectively. As an example, audio file name 'ID31_hc_0_1_1' depicts 31st person who is a healthy control with H&Y rating of 0, UPDRSII-5 rating of 1 and UPDRSIII-18 rating of 1. Parkinson's disease and voice loss are related but not identical, because of that this audio file represents HC person whose UPDRS rating is more than 0.

In this chapter, Healthy Controls (HC) were represented by binary label '1' and PD patients were represented by binary label '0'. It is a binary classification problem. Sampling rate and bit depth of each audio signal is 44.1 kHz and 16 Bit respectively. From each audio recording, multiple 10-second clips have been taken for classification. Total 229 audio clips (128 audio clips of HC and 101 audio clips of PD) with 10 second duration were used for this analysis. Among 229 audio clips, 80% audio (102 audio clips of HC and 81 PD patient's audio clips) have been taken as training data and remaining 20% audio (26 audio clips of HC and 20 PD patient's audio clips) have been taken as testing data.

2.4 METHODOLOGY

In this section, data pre-processing steps and different classifiers which were used for the classification have been discussed. Data pre-processing consists of mainly two steps, namely signal processing and features extraction. For features extraction two signal processing techniques, namely Wavelet Transform and Cross-wavelet Transform along with transfer learning architectures (VGG16, VGG19, DenseNet-121 and Xception) have been deployed. In the classification part, different ML algorithms namely SVM, KNN, Random Forest, Logistic Regression and majority voting ensemble have been mentioned.

2.4.1 DATA PRE-PROCESSING

Data pre-processing contains mainly two steps,

a) Signal Processing,

b) Features Extraction.

2.4.1.1 SIGNAL PROCESSING

Two Signal processing techniques have been used for extracting time-frequency scalograms from recorded audio signals, namely (i) Wavelet Transform, (ii) Cross-wavelet transform.

2.4.1.1.1 CONTINUOUS WAVELET TRANSFORM

Wavelet Transform overcomes the limitations of Fourier Transform[32]. Wavelet Transform provides better result for non-stationary signals than the traditionally used Fourier Transform. In non-stationary signals, for a certain time-duration, frequencies are varied with time. Fourier Transform (FT) is only appropriate for stationary signals where frequency domain information is required. Time information of the signal is lost because it converts the signal from time-domain to frequency-domain, which is ideal for stationary signals. But for non-stationary signals, it is not ideal because time domain information is required along with frequency-domain representation. Without time-domain information, it is hard to determine when a certain event occurred.

Audio signals are mainly non-stationary signals [33]. The STFT was developed to address Fourier Transform's poor time resolution problem which provides time-frequency scalograms of the signal. A portion of a non-stationary signal can be presumed to be stationary using STFT. Then a fixed length of window function is taken and move it along the signal from start to end and take a Fourier Transform at each stationary section. Time and frequency resolutions are constant because the window length is fixed for entire length of the signal. The time resolution of a narrow window is good, but the frequency resolution is low, while a wider window has poor

time resolution but good frequency resolution. To overcome this problem Wavelet Transform was introduced.

Multi-resolution analysis is the result of using the Wavelet Transform (WT) to break down a signal into multiple frequencies at different time resolutions.. At high frequencies, it has good time resolution but poor frequency resolution. However, frequency resolution is good at low frequencies but time resolution is poor. Wavelet works as a window function in Wavelet Transform. The wavelet's width and central frequency can be altered, and it can move across the signal by changing scale (1/frequency). The expanded wavelet is more effective at resolving low frequency signal components with poor time resolution (large values of scaling). Shurnken wavelet has a superior time resolution and is better at resolving high frequency components of the signal (small values of scaling). The Wavelet Transform is defined as:

$$W(s,\tau) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{|\sqrt{s}|}} \Psi^* \left(\frac{t-\tau}{s}\right) dt \qquad (2.1)$$

$W(\boldsymbol{s},\boldsymbol{\tau})$ is the continuous wavelet transform of the audio signal, generated from a mother wavelet Ψ(t). Scale factor and translation factors are represented by 's' and 'τ'.

Mother wavelet is a simple oscillatory function of finite duration that integrates to zero and is square integrable, which means it has finite energy and meets the admissibility condition. Other window functions can be generated by using the mother wavelet. Different types of mother wavelets are present namely,

(a) Symlet3,

(b) Daubechies3,

(c) Morlet,

(d) Mexican Hat,

(e) Meyer

The admissibility and regularity conditions are the most significant qualities of wavelets, and these are the properties that gave wavelets their name. Admissibility criterion is defined as,

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty \tag{2.2}$$

The Fourier Transform of Ψ(t) is Ψ(ω). According to this condition, Ψ(ω) disappears at zero frequency, i.e.

$$|\Psi(\omega)|^2{}_{\omega=0} = 0 \tag{2.3}$$

This implies that wavelets must have a spectrum that is similar to that of a band-pass filter.

As the DC gain of the wavelets at zero frequency is zero, so the wavelet's average value in time domain must be zero,

$$\int \Psi(t) dt = 0 \tag{2.4}$$

As a result, it is a oscillatory function which implies that mother wavelet is a wave.

In both the temporal and frequency domains, the wavelet function should have some smoothness and concentration, according to the regularity criterion. It states that wavelet transform decreases quickly with decreasing scale. These both conditions together provide the wavelet transform.

Figure 2.1 represents the continuous wavelet transform (wavelet scalogram) of a recorded audio signal.

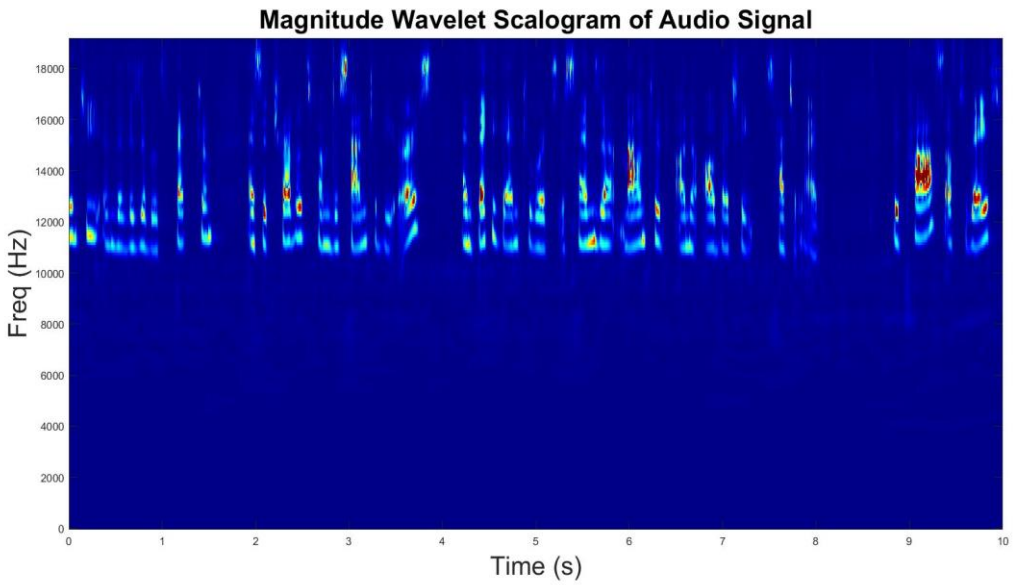


Fig 2.1. Magnitude spectrum of wavelet transform of an audio signal

2.4.1.1.2 CROSS-WAVELET TRANSFORM

The XWT is the Hadamard product of the wavelet transform matrix of the two signals [34]. The XWT is defined as follows if x(t) and y(t) are two time-domain signals:

$$W^{xy}(s,\tau) = W^{x}(s,\tau)W^{y*}(s,\tau) \qquad (2.5)$$

$W^{x}(s,\tau)$ and $W^{y}(s,\tau)$ are the wavelet transforms matrix of two signals respectively. 's' and 'τ' are scale factor and translation factors. Cross-wavelet scalogram shows regions in time-frequency plane where two waveforms have high common power. One audio signal has been considered as the reference signal. Then Cross-wavelet transform has been applied between the reference signal and other audio signals. Each audio signal including reference signal has been taken as 10 seconds of time duration. Reference audio signal was considered as x(t), while other audio signals were defined as y(t). In this chapter, audio signal of a HC was taken as reference signal. Morlet wavelet is taken as mother wavelet for the cross-wavelet transform. This choice of mother wavelet is solely based on the problem's nature. Magnitude of $\boldsymbol{W^{xy}(s,\tau)}$ and

phase angle ($\emptyset$) is defined as $tan^{-1}\frac{I\{w^{xy}\}}{R\{W^{xy}\}}$. R$\{w^{xy}\}$ and I$\{w^{xy}\}$ are real and imaginary value of the Cross-wavelet spectrum respectively. Figure 2.2 shows the Cross-wavelet scalogram between the audio signal of HC and PD patient. The Cross-wavelet spectrum between HC and HC is shown by figure 2.3.

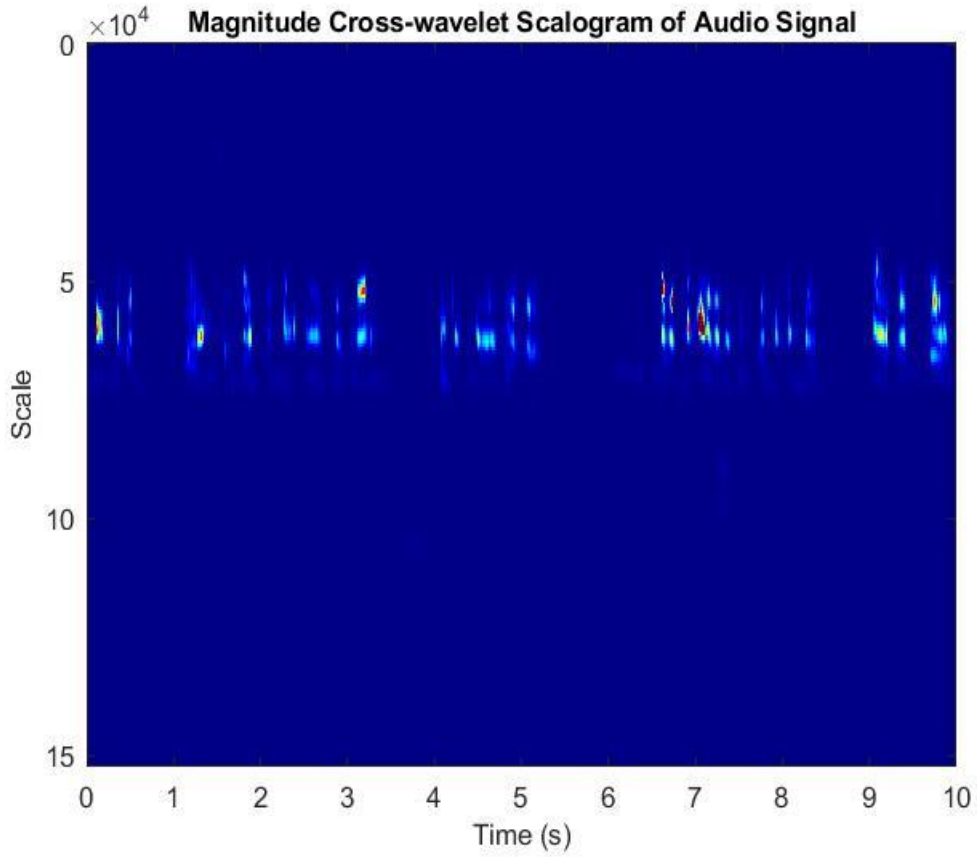


Fig 2.2. Cross-wavelet spectrum between HC and PD patient

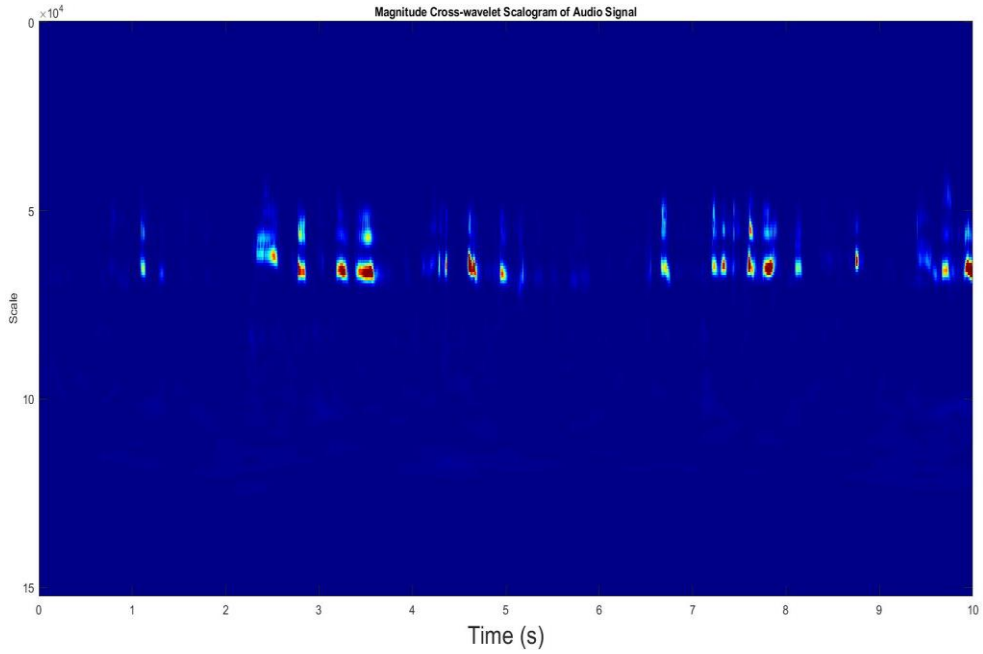


Fig 2.3. Cross-wavelet spectrum between HC and HC

2.4.1.2 FEATURES EXTRACTION USING TRANSFER LEARNING

Feature extraction is one of the most important step in ML and DL classification problem. Extracted features are then fed into different ML algorithms for the classification. Feature extraction is done by converting raw signal (audio, image, video) into the numerical features while maintaining the raw original data set's metadata. Features can be extracted manually or automatically by using deep learning models from original signals. Different time domain and statistical (Root mean square, kurtosis, skewness, crest factor, form factor, maximum and minimum value), frequency domain (Fast Fourier Transform coefficients, mean frequency, signal-to-noise ratio, spectral moment, energy) and time-frequency domain (Mel frequency cepstral coefficients, wavelet, cross-wavelet) features are extracted manually. In this chapter, pre-trained DL models such as VGG16, VGG19, DenseNet-121 and Exception have been used to extract features from scalogram images.

2.4.1.2.1 VGG16 MODEL FOR FEATURES EXTRACTION

VGG16 is one of the most preferred Convolutional Neural Network (CNN) architecture and it is trained on ImageNet database which contains more than one million images [35]. It has 13 convolutional layers along with 3 fully-connected layers [36]. In this chapter, 3 fully connected layers have been dropped and used the remaining model as feature extractor. All the convolution layers in VGG16 are having a kernel size of 3 x 3, stride of 1 and padding is 'same'. Max-pooling layer is used after every stack of convolutional layers which has a kernel size of 2 x 2 and stride of 2.

VGG16 model has been used for features extraction from the wavelet and cross-wavelet scalograms. These time-frequency scalograms are a fixed-sized 224 x 224 x 3 RGB image. Image normalization was done by dividing every pixel of the RGB scalogram images by 255. Then each normalized image was passed through 2 convolutional layers which contains 64 filters followed by ReLU activation function. Then the activation map was passed through the max-pooling layer. The shape of the activation at the output of the max-pooling layer was 112 x 112 x 64. Then it was passed through the second stack of convolutional layers and one max-pooling layer. Convolutional layers in the second stack contain 128 filters. The shape after the max-pooling layer was 56 x 56 x128. This was followed by the third stack with three convolutional layers of 256 filters and one max-pooling layer. The output shape was 28 x 28 x256 . Then it was passed through two stacks of three convolutional layers and each stack has one max-pooling layer.Convolutional layers of each stack contains 512 filters and the output at the end of was 7 x 7 x 512.

Then the activation map of size (7 x 7 x 512) was flatten and at the output we got a feature vector of size (7 x 7 x 512= 25088) features. This vector was used as the feature vector for Parkinson's disease classification. For each scalogram RGB image, 25088 features were present.

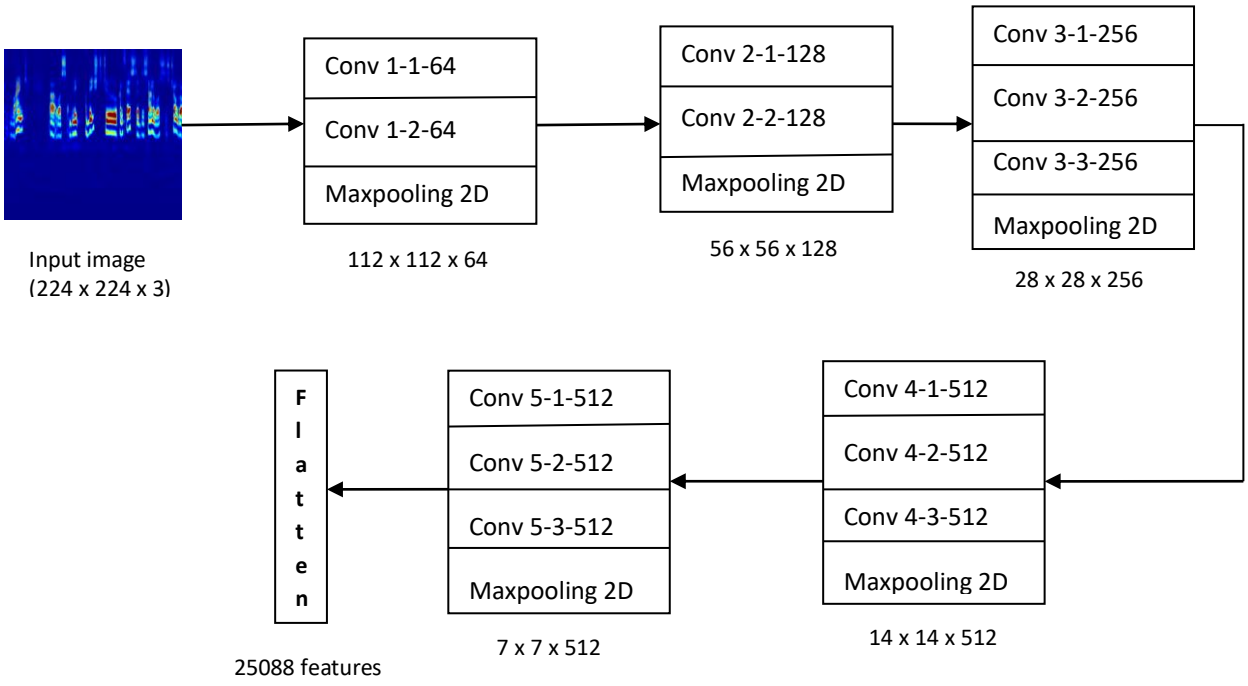


Fig. 2.4. Features extraction using VGG16 from Wavelet and Cross-wavelet Scalograms

In this figure, Wavelet scalogram has been used for feature extraction using VGG16. In the similar way, features were extracted from the Cross-wavelet scalogram images.

2.4.1.2.1.1 CONVOLUTION

The mathematical combining of two functions to generate a third function is referred to as "convolution." When this occurs, two pieces of data are combined.

In CNN, convolution is conducted on the input data using a filter or kernel to build a feature map. Convolution is performed by moving the filter over input and a matrix multiplication is performed at each position, and the result is added to the feature map.

Suppose a kernel is taken for convolution operation with 3 x 3 shape. Now the element wise multiplication is done between the kernel and a 3x3 sized area of the input image's matrix. Then the summation is done over this section to get the one element of output feature map. This convolution is taken place repetitively by

moving the kernel over input image and output feature map is occurred. This feature map contains important information about the image.

An RGB image with shape of n x n x 3 is considered. Kernel size is taken as k x k. Then the shape of feature map is given as:

$$f = \frac{n - k + 2p}{s} + 1 \qquad (2.6)$$

f x f x 1 is the shape of the feature map. Strides ('s') is defined as how many pixels the filter will shift. Padding ('p') is also a important component in CNN. Two types of padding is present, 'valid' padding and 'same' padding. In valid padding, no padding is occurred and input image is same as it was. In 'same' padding, zeros are padded at the outside of input image matrix in such a way that the shape of input image and feature map is same. Multiple filters are used to improve the feature map's depth.

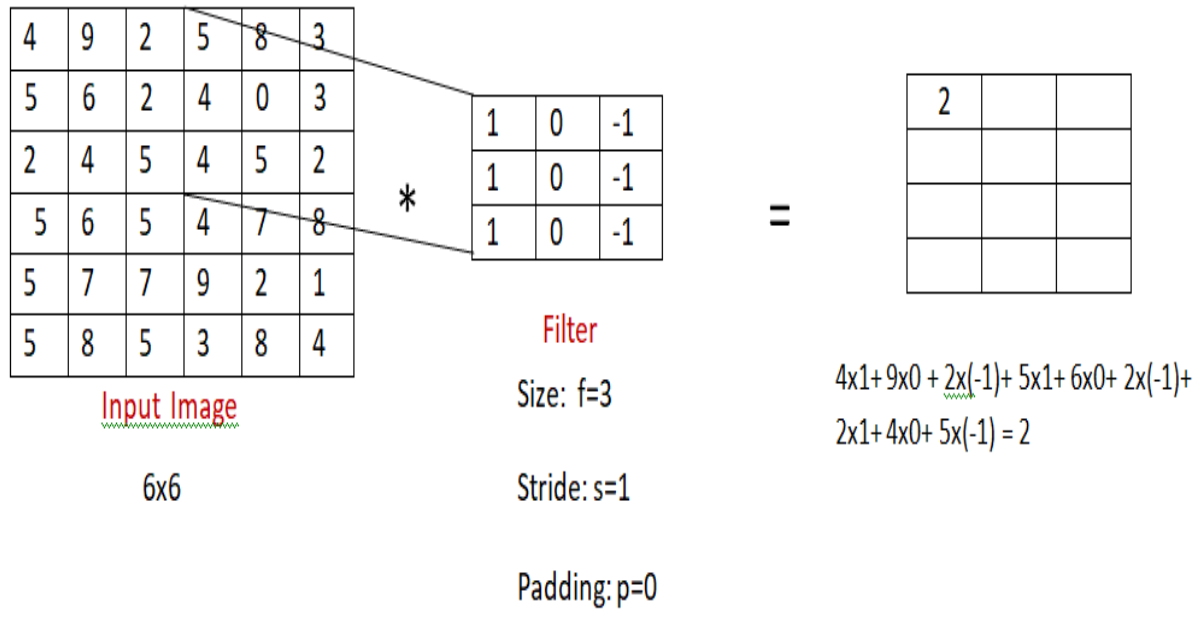


Fig 2.5. Convolution operation

2.4.1.2.1.2 POOLING LAYERS

Pooling layers reduce the size of feature map. Because of pooling layers width and height of the feature map reduces but the depth of the feature map remains same. It reduces the computational complexity to a certain extent in the process of obtaining the most important features from feature maps.

Two types of pooling layers are present in the study, namely max pooling and average pooling. A pooling layer with shape of 2 x 2 is considered. Max pooling results the maximum value of the pixels in the part of the image covered by the 2 x 2 layer. Average pooling provides the average value of that image's portion. Max pooling is better than average pooling in terms of extracting dominant features. Figure 2.6 depicts the max pooling and average pooling operation on the feature map.

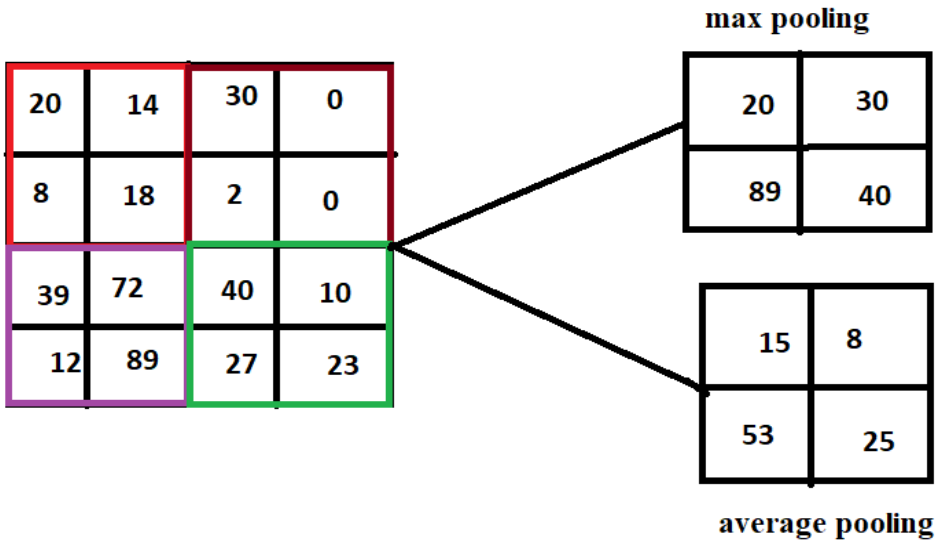


Fig. 2.6. Max pooling and Average pooling operation

2.4.1.2.1.2 RELU ACTIVATION FUNCTION

An activation function determines whether or not a neuron should be fired or activated. It causes a neuron's output to become nonlinear.

In CNN, Rectified Linear Unit (ReLU) is the most prevalent activation function, which is expressed as: $\boldsymbol{f(x) = \max(0, x)}$. If x is greater than zero then the output produces x, whereas x is less than zero then output will be zero. Figure 2.7 introduces the ReLU activation function.

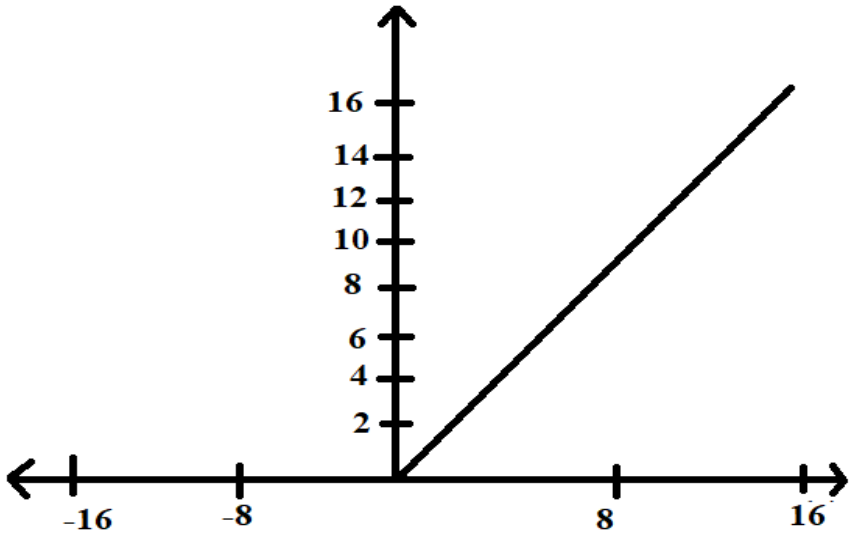


Fig 2.7. ReLU activation function

Since the mathematical process is simpler and the activation is sparser, ReLU is not computationally expensive than certain other typical activation functions like tanh and Sigmoid. If the input x is less than zero, there's a good probability that a certain unit won't fire at all. Sparsity also reduces noise and overfitting, as well as more compact models with higher predictive value. Neurons in a sparse network process significant information to the next layer. As an example, if a neuron which is responsible for identifying a certain object in a object classification model should not be triggered if the image is human face.

Another benefit that ReLU has is that it converges faster. When x>0, slope of the ReLU is one and it does not increase as x increases. As a result, ReLu does not suffer from the vanishing gradient problem like the other activation functions, such as Sigmoid or tanh.

In this chapter, ReLU activation function has been introduced after each convolution layer.

2.4.1.2.2 VGG19 MODEL FOR FEATURES EXTRACTION

VGG19 is a CNN architecture which was proposed by [37]. This architecture consists 16 convolutional layers along with 3 fully-connected layers and it is also trained on ImageNet database. Last three fully connected layers have been dropped and remaining 16 convolutional layers have been used as the feature extractor. All the convolution layers in VGG19 are having a kernel size of 3 x 3, stride is 1 with padding is 'same' followed by ReLU activation function. Then Max-pooling layer is used after every stack of convolutional layers with a kernel size of 2 x 2 and stride of 2.

Wavelet and Cross-wavelet scalogram images (RGB) of size 224 x 224 x 3 were taken as the input to the VGG19 feature extractor. Then image normalization was done similarly as it has been mentioned in section 2.4.1.3.1. Then each normalized image was passed through first stack of 2 convolutional layers which contains 64 filters followed by ReLU activation function. Then the activation map was passed through the max-pooling layer. The shape of the activation at the output of the max-pooling layer was 112 x 112 x 64. Then it was passed through the second stack of 2 convolutional layers and one max-pooling layer. Each convolutional layer in the second stack contains 128 filters. The shape after the max-pooling layer was 56 x 56 x 128. This was followed by the third stack with four convolutional layers of 256 filters and one max-pooling layer. The output shape was 28 x 28 x 256 . Then it was passed through two stacks of four convolutional layers and each stack has one max-pooling layer. Convolutional layers of fourth and fifth stack contains 512 filters. The output at the end of the feature extractor was 7 x 7 x 512.

The activation map of size (7 x 7 x 512) was then flattened, yielding a feature vector of size (7 x 7 x 512) 25088 features as the output. This vector was utilised as a feature vector in the classification of Parkinson's disease.

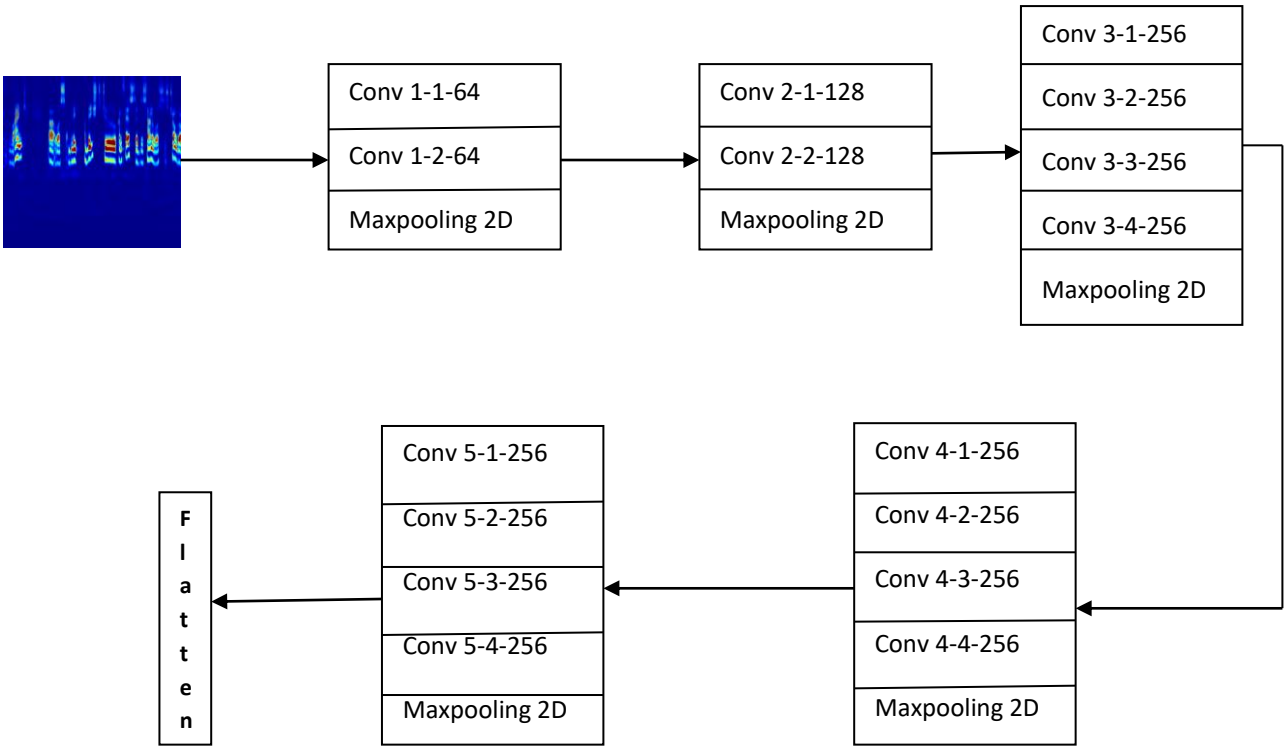


Fig. 2.8. Features extraction using VGG19 from Wavelet and Cross-wavelet Scalograms

2.4.1.2.3 DenseNet-121 MODEL FOR FEATURES EXTRACTION

DenseNet-121 architecture is a dense CNN model which was proposed by (Huang & Liu, 2017) [38]. This architecture consists of four dense blocks, three transition layers, convolution and max-pooling layer along with fully connected layers. Fully connected layers have been dropped for feature extraction.

In DenseNet-121 architecture, every normalized Wavelet and Cross-wavelet scalogram image of size 224 x 224 x 3 has been passed through a convolutional layer which has kernel size of 7 x 7 and stride is 2 with 64 filters. The resulting image had a resolution of 112 x 112. Then it was passed through a max-pooling layer which had a kernel size of 3 x 3 and stride 2. So the image's resolution was again reduced by half (56 x 56). Then it was passed through four dense blocks and between two dense blocks one transition layer was present. First dense block was made up of two convolutional layers that have been repeated by 6 times. One

convolutional layer had 3 x 3 kernel and other had a kernel size of 1 x 1 with 'same' padding and stride 1. The second, third, and fourth dense blocks were similarly built up of two convolutional layers that were repeated 12, 24, and 16 times. Each transition layer had one convolutional layer with a kernel size of 1 x 1 and one average pool layer with kernel size of 2 x 2 and stride of 2. So after every transition layer resolution of the image was decreased by two. Figure 2.9 represents features extraction using DenseNet-121 architecture.

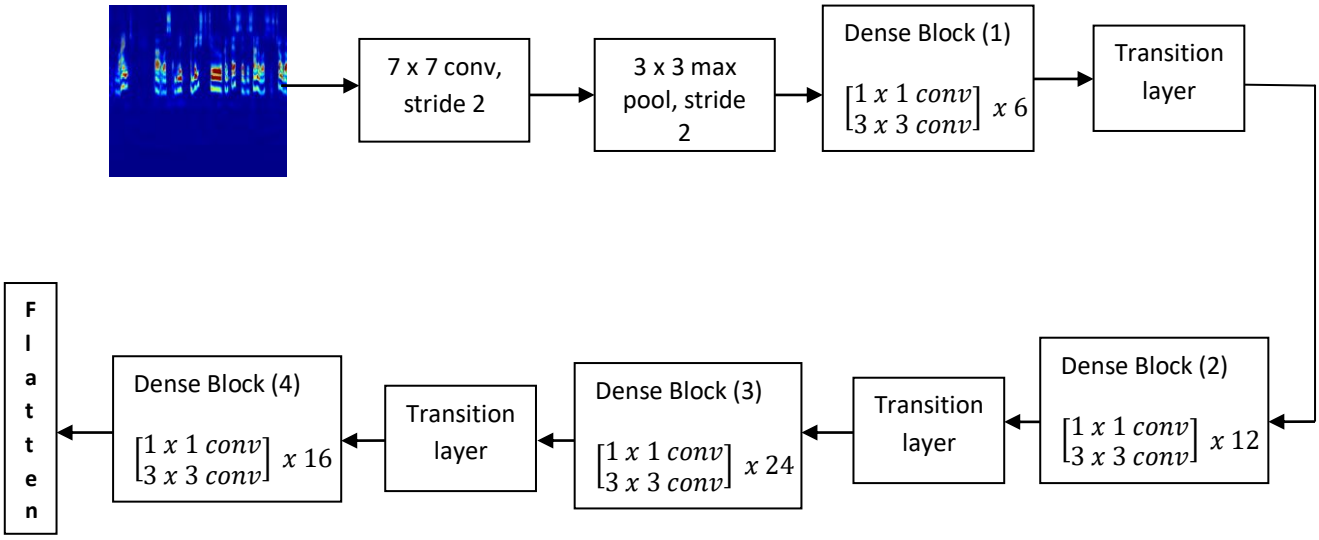


Fig. 2.9. Features extraction using DenseNet-121 from Wavelet and Cross-wavelet Scalograms

Figure 2.10 shows the architecture of dense block. In dense block all the dense units were directly connected with the subsequent dense units and newly generated features were also passed to subsequent dense units. So the shallow features of previous dense units were reused and efficiently employed again and again. Vanishing gradient problem with DenseNet-121 architecture can be reduced to a certain extent. Between two layers in dense block, one bottleneck layer with kernel size of 1 x 1 and one convolutional layer with kernel size of 3 x 3 for convolution operation were present. Each layer has been followed by batch normalization (BN) and ReLU activation.

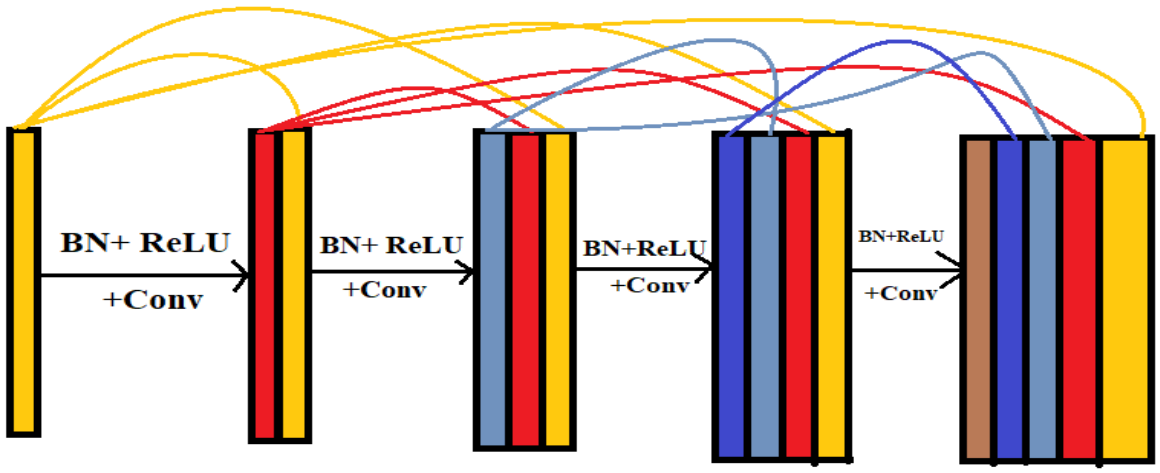


Fig. 2.10. Diagrammatic representation of dense block

Figure 2.11 shows the structure of transition layer.

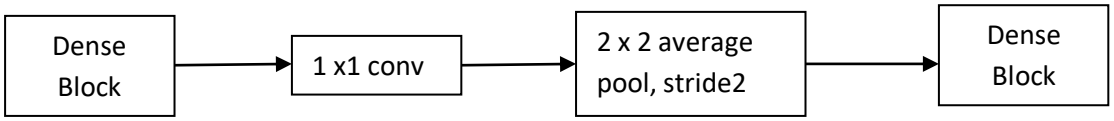


Fig. 2.11. Transition layer

2.4.1.2.4 XCEPTION MODEL FOR FEATURES EXTRACTION

Xception is a transfer learning based CNN model which has been trained on ImageNet database. This architecture was proposed by (Chollet F.)[39]. It is an extreme version of Inception V3 model which is used depth-wise separable convolution rather than conventional convolution process. In depth-wise separable convolution, two convolution processes are present, namely depth-wise convolution and point-wise convolution instead of one single convolution. In depth-wise convolution, convolution is done at every channel of the original image. Suppose an RGB image is taken which has 3 channels. A filter with kernel size of 3 x 3 is considered for the depth-wise convolution operation. Now for performing convolution operation, 3 individual 3 x 3

filter is required. Point-wise convolution is done by 1 x 1 convolution to compress the output of depth-wise convolution over the depth. This architecture reduces the computational complexity. In Inception V3 model point-wise convolution is used at first on the input image and then depth-wise convolution is used on each of the depth spaces from each of those input spaces. Xception model simply reverses this process. It first applies depth-wise convolution and then point-wise convolution is applied. Between these two convolutions, no non-linear activation function is introduced. Figure 2.12 demonstrates the architecture of Xception model.

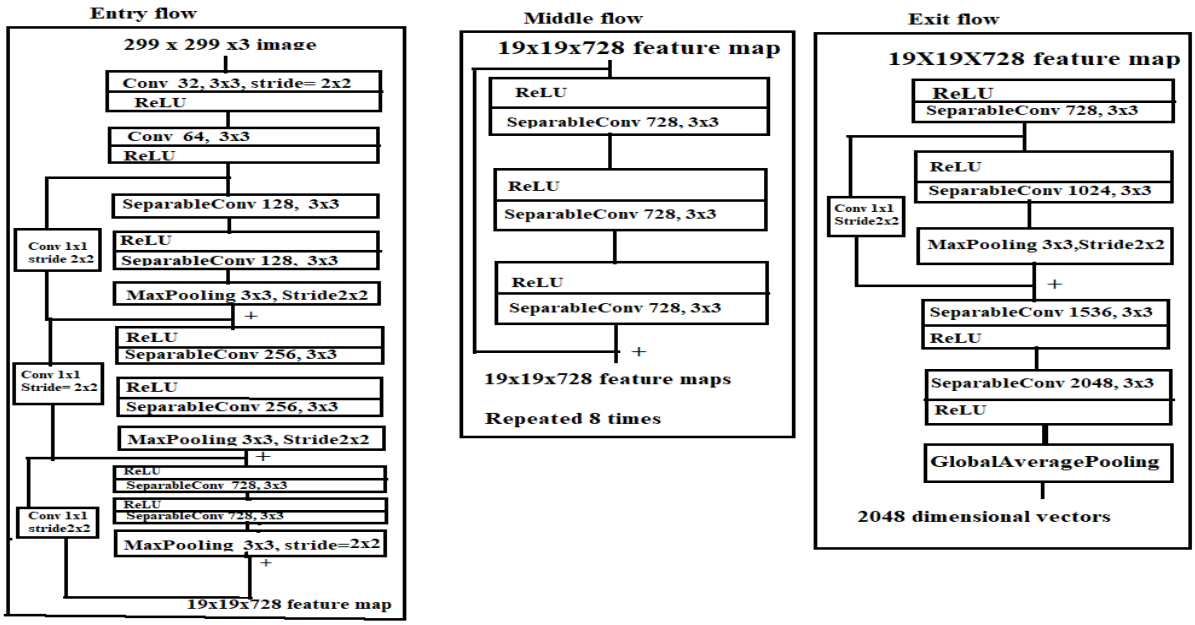


Fig. 2.12. Exception model architecture

For feature extraction using this model, Wavelet and Cross-wavelet scalogram images of shape 224 x 224 x 3 has been reshaped to 299 x 299 x 3. Then each image has been passed through entry flow, middle flow and exit flow as shown in the figure. The middle flow was repeated for 8 times. Fully connected layers and Logistic Regression classifier have been dropped for extracting the features. A vector size of 2048 dimensions was then fed to the different ML models for classification.

2.4.2 CLASSIFICATION

Extracted deep features from the Wavelet and Cross-wavelet scalogram images using VGG16, VGG19, DenseNet-121 and Xception model were then feed into different ML algorithms, such as Random Forest, Logistic Regression, SVM and KNN. Then Majority Voting Ensembling technique has been introduced on these models to improve the evaluation metrics (accuracy, precision, recall, F1-scoe and AUC score). All the above mentioned ML algorithms have been discussed in this section.

2.4.2.1 RANDOM FOREST

Random Forests is a well-known classification and regression algorithm that is capable of effectively categorizing large datasets. An ensemble of decision trees is generated via the Random Forests method. The primary premise of ensemble methods is to put weak learners together in order to create a strong learner [40]. The data is inputted at the top of the tree, and as it descends, it is sampled at random, but with replacing into successively smaller sets. Random Forest selects base features from a random set of features. It generates highly uncorrelated classifiers and trades variance with bias. The following steps can be used to implement the RF algorithm.

1. For b=1 to B :

(i) From the training data, create a bootstrap sample Z* of size N.

(ii) The next steps should be repeated iteratively for each terminal node of the tree until the minimal node size n_{min} is attained in order to grow a random forest tree T_b.

(a) From the p variables, choose m variables at random.

(b) Choose from the m, the best variable or splitting attribute.

(c) Create two daughter nodes by splitting the node.

2. produce the ensemble of trees$\{\boldsymbol{T_b}\}_{\boldsymbol{1}}^{\boldsymbol{B}}$.

B is the total number of decision trees. Total m training data points are available and from which p variables are taken as bootstrapped data.

2.4.2.2 SUPPORT VECTOR MACHINE

The support vector Machine (SVM) are arguably the most well-known supervised learning algorithm which is used for both classification and regression problem and it is specialised for a lower number of training samples. SVM was proposed by VN Vapnik [41]. The basic principle of SVM is to identify an optimal hyperplane in the feature plane which acts as a decision boundary. This decision boundary separates the data points between different in the feature plane. It tries to put a linear boundary between the two classes, indicated by a bold line, and orients it so that the margin is maximised, i.e. the separation in between boundary and the closest input data in each class is maximum.

Let's consider a binary classification problem which has two class labels such as class1 and class2. Now m data points with n features are taken which are divided between class1 and class2. Input vector can be expressed as $X_j = [x_{1j}, x_{2j}, x_{3j,\ldots..}, x_{nj}]$, and j is varied from 1 to m. Output can be expressed as $y^{(j)} = \{-1,1\}$. Consider the data points are linearly separable. So SVM tries to put an optimal hyper plane in the n-dimensional feature plane which divides the data points between two classes. Two classes are separated by that hyperplane. The separating hyper plane can be described as:

$$g_\theta = \theta^T x + \theta_0 \tag{2.7}$$

Where $\theta \in R^n$ is a weight vector and θ_0 is bias. These two parameters determine the separating hyper plane between two classes.

Margin in SVM is an unsigned distance which defines the smallest distance between a data point $x^{(j)}$ and a separating hyper plane H as

$$\gamma = min_{j=1,2,..,m}\, d(x^j, H) \tag{2.8}$$

$\boldsymbol{d(x^j, H)}$ is the unsigned distance between j-th point and the hyper plane. γ defines the margin value. $\boldsymbol{d(x^j, H)}$ can be expressed as:

$$d(x^j, H) = \frac{|g_\theta(x^{(j)})|}{\|\theta\|_2} \tag{2.9}$$

The signed distance between the data points and separating hyper plane can be expressed as:

$$d_{signed}(x^j, H) = y^{(j)} \left(\frac{\theta^T x^j + \theta_0}{\|\theta\|_2}\right) \tag{2.10}$$

If $\boldsymbol{d_{signed}(x^j, H)} \geq 0$, SVM correctly classifies $x^{(j)}$, otherwise it misclassifies the data.

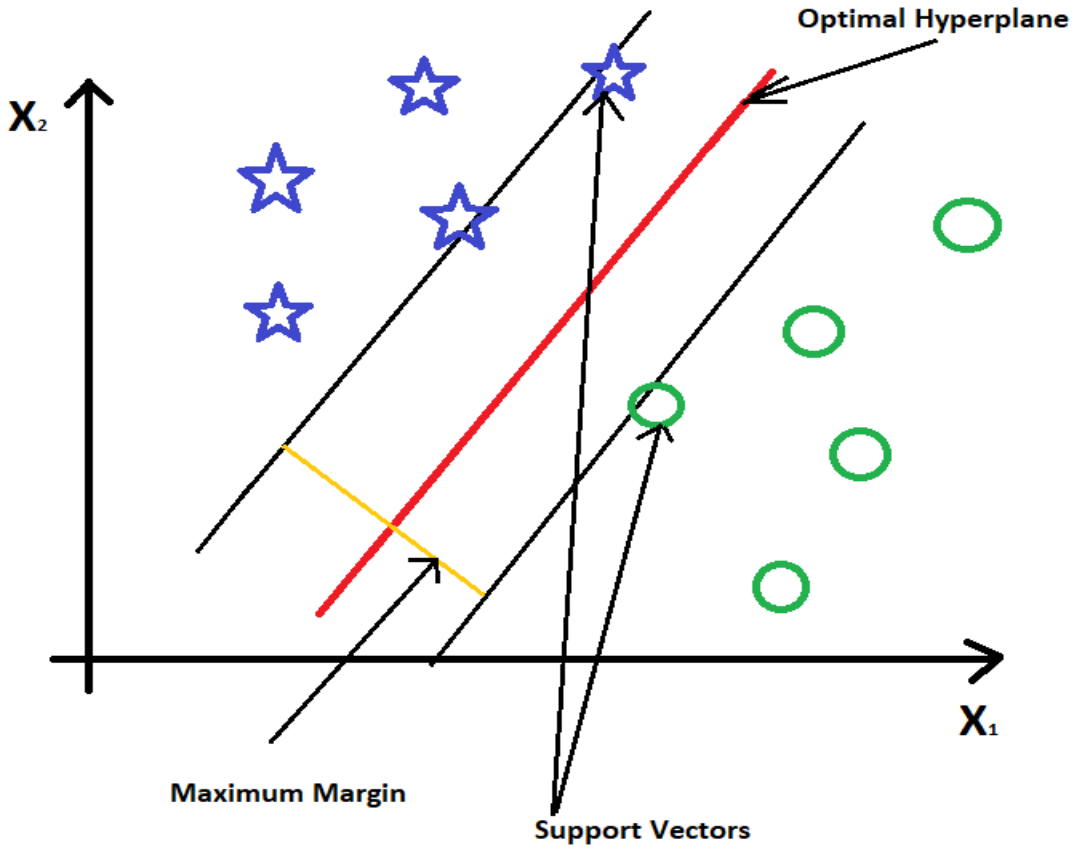


Fig 2.13. The optimized hyperplane in SVM

SVM tries to maximize the margin, so the optimization problem can be given as:

maximize γ

subject to $y^{(j)}\left(\frac{\theta^T x^j+\theta_0}{\|\theta\|_2}\right) \geq \gamma$ (2.11)

This optimization problem is not easy because γ depends upon θ and θ_0 and $\frac{\mathbf{1}}{\|\boldsymbol{\theta}\|_2}$ is non-linear.

So this problem can be generalized as:

minimize $\frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{\theta}\|^{\mathbf{2}}$

subject to $y^j(\theta^T x^j + \theta_0) \geq 1$ (2.12)

The data points which are closer to the separating hyper plane, known as support vectors. Support vectors have an impact on the position and orientation. The margin between two classes can be maximized using support vectors. If the data points are not linearly separable, then the classification problem becomes non-linear and a linear separable hyper plane is insufficient to adequately distinguish the two classes. The classification features are generated from the original data using nonlinear mapping. So for nonlinear classification problem, kernel functions transform data points on a high dimensional feature plane where the data points are linearly separable. Some kernel functions which are used in SVM are linear functions, radial basis functions and polynomial functions.

Radial Basis function can be expressed as:

$$k(x1, x2) = e^{-\frac{\|x1-x2\|^2}{2\sigma^2}} \quad (2.13)$$

Where σ^2 is the width. C and σ^2, these two parameters influence evaluation metrics of the classifier. In this chapter RBF kernel has been used for classification.

2.4.2.3 LOGISTIC REGRESSION

A supervised machine learning approach called logistic regression is employed for classification problems and it evaluates the relationship between labels (dependent variables) and one or more independent features. Let's consider a binary classification problem with m input data points which have two independent features. Binary classification represents two class labels 1 and 0, where Y ∈ {0,1}. Input vector can be expressed as $\vec{x}$ = {$(x_1^{(1)}, x_2^{(1)})$, $(x_1^{(2)}, x_2^{(2)})$,…….,$(x_1^{(m)}, x_2^{(m)})$} and output vector is $\vec{y}$ = {$y^{(1)}$, $y^{(2)}$, $y^{(3)}$, $y^{(4)}$,….., $y^{(m)}$}. Logistic Regression's job is to predict the output either 0 or 1. Now Linear Regression tries to fit an optimal usually least-squares fit to some given data using weights which are linear. Output of simple Linear Regression model can be expressed as:

$$z = w_0 + w_1 x_1 + w_2 x_2 \tag{2.14}$$

Now for classification problem, all data have to be squished between 0 and 1. For that purpose, sigmoid function (σ) is used. Output of Linear Regression model is fed to the sigmoid function which can be expressed as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{2.15}$$

As z tends to ∞, σ(z) tends to 1 and if z tends to –∞, σ(z) tends to 0. At z=0, σ(z) =0.5. This is also a monotonic function.

Figure 2.14 represents the sigmoid activation function.

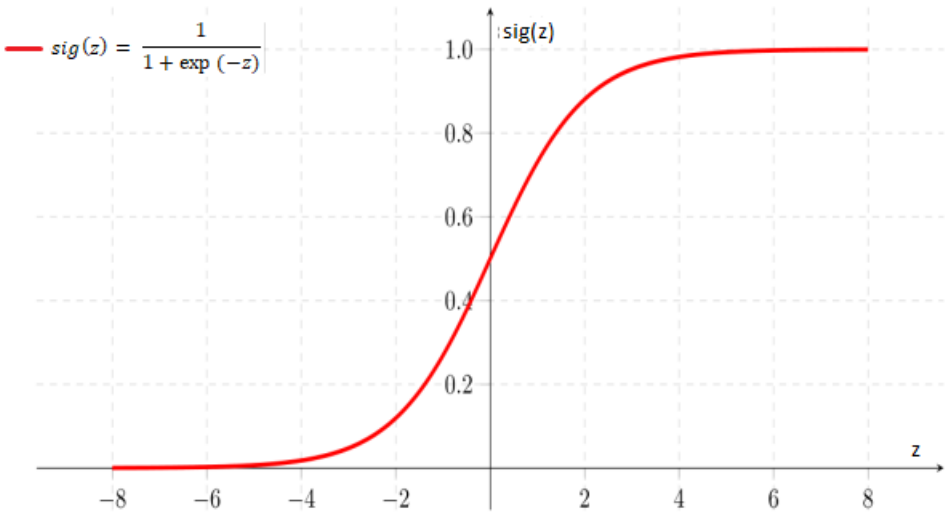


Fig. 2.14. Sigmoid Activation Function

Predicted output from the Logistic Regression can be given as:

$$\hat{y} = \sigma(w_0 + w_1 x_1 + w_2 x_2) \tag{2.16}$$

Where $\hat{y} \in [0,1]$. $\hat{y}$ can also be interpreted as p(y=1|x). It defines that the probability of y is label 1 for a given input data.

For $z > 0$, $\hat{y} > 0.5$. Then the point is labeled as class 1. It also means that probability of that point to present in class 1 is greater than 0.5.

For $z < 0$, $\hat{y} < 0.5$. Then and the probability of that point to present in class 1 is less than 0.5. So the point is labelled as class 0.

2.4.2.4 K-NEAREST NEIGHBORS

The k-Nearest Neighbors (KNN) is a non-parametric, supervised learning algorithm which is used for both classification and regression problem. This method does not make any assumption about the mapping function between the

output dependent variables and input independent features. In the feature space, the input consists of the K closest training instances. It uses majority voting to get more accurate result.

For an example, a binary classification problem is considered where two classes (0 or 1) are present and k is assumed as 3. k represents how many nearest neighbors have to consider. Now a new test point is taken into the feature space. Three data points have to find out from the feature space those are nearer to the new test point based on eucledian or manhattan distance. Among these three points, if two points are coming from the label 1 and one point is from the label 0, then the new point is considered as the label 1.

2.4.2.5 MAJORITY VOTING CLASSIFIER

Majority voting classifier is an ensemble ML algorithm that incorporates predictions from several different machine learning models. This algorithm might be utilized to improve model performance with the intention of exceeding any individual model in the ensemble. In this technique, predictions from many machine learning models are combined in a voting ensemble. It can be applied to problems involving classification and regression. It can be used for both classification and regression problem. In regression problem, predictions from all the machine learning models are taken as average. In classification problem, predictions are taken from the majority vote of all machine learning models.

Majority voting classifiers are of two types, such as Hard Voting and Soft Voting. Hard voting entails adding up all of the predictions for each class label and predicting the one with the most votes. Soft voting entails adding up the anticipated probabilities for each class label and predicting the one with the highest likelihood.

2.4.3 FLOWCHART OF THE PROPOSED METHODOLOGY

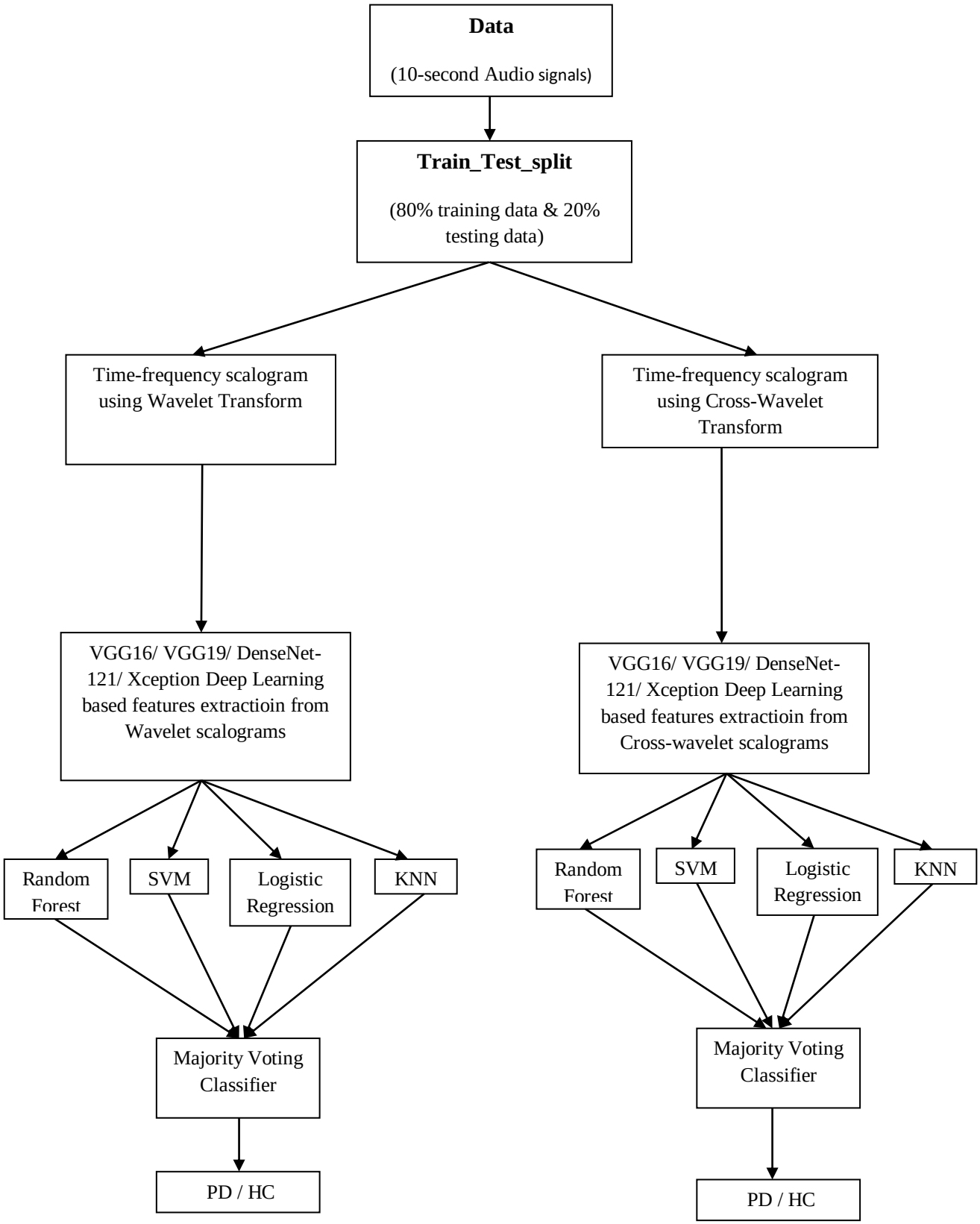


Fig. 2.15. Flowchart of proposed methodology

2.5 EXPERIMENTAL RESULTS

In the previous section proposed methodology (data) of the Parkinson's disease classification has been discussed. In this section, accuracy, recall, precision, f1-score and area under curve (AUC) were used to evaluate the performance of different ML classifiers.

Confusion Matrix :

Confusion matrix is table which a way of evaluating how well a machine learning algorithm performs in a classification problem when the output can include two or more classes. It is a matrix which includes four different parameters of true and predicted class labels.

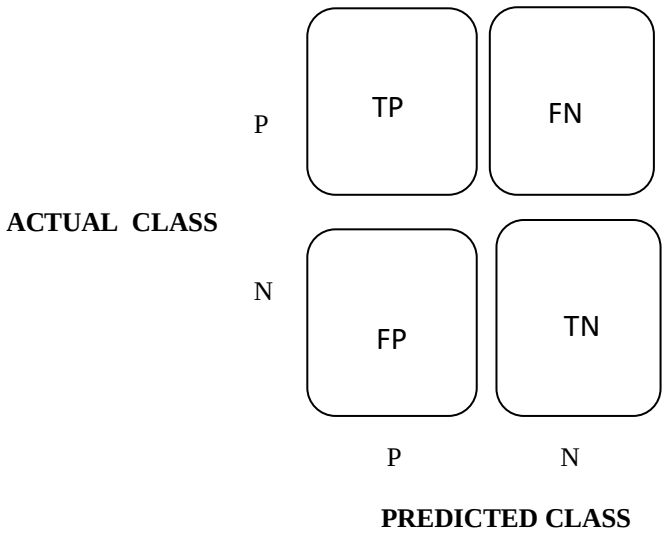


Fig 2.16. Confusion Matrix

where,

TP : True Positive, **TN :** True Negative, **FP:** False Positive, **FN:** False Negative

In this chapter, PD was taken as P (Positive) and HC was taken as N (Negative).

True Positive :

If the predicted class label by ML classifiers is positive, which is same as the true class label, then it is known as True Positive (TP). For PD classification, TP

represents that the person has Parkinson's disease and machine learning classifier also predicts the same.

True Negative :

These are the accurately predicted negative class labels, indicating that neither the actual class label nor the predicted class labels are positive. E.g. If the predicted class and true class tell the same thing that the person has the Parkinson's disease, then the outcome is True Negative (TN).

False Positive:

It is a result when the model forecasts negative class inaccurately. E.g. If the ML classifiers predict class label as PD but the true class label is HC, then the result is False Positive (FP). FP should be less as it decreases the performance of classifiers.

False Negative:

It is a result when the model forecasts positive class inaccurately. E.g. If the ML classifiers predict class label as HC but the true class label is PD, then the result is False Positive (FP). FP should be less as it also decreases the performance of classifiers.

The evaluation metrics are given as follows:

Accuracy:

The easiest performance metric to understand is accuracy, which is just the proportion of properly predicted observations to all observations. Accuracy is an excellent indicator, but only when the total number of the false positive and false negative are nearly equal in the datasets.

$$\boldsymbol{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{2.17}$$

Precision:

It is the percentage of correct positive classifications (True Positives) from cases that

are predicted as positive. E.g. It shows how many people are accurately identified as having Parkinson's disease out of all the predicted PD patients.

$$\boldsymbol{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.18}$$

Recall:

It is the percentage of correct positive predicting classifications (true positives) from cases that are truly positive. E.g. It shows how many people are accurately identified as having Parkinson's disease out of all the actual PD patients.

$$\boldsymbol{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.19}$$

f1-score:

By calculating a classifier's harmonic mean, the F1-score combines its precision and recall into a single metric. f1-score is a better evaluation metric than accuracy when the data set distribution is uneven (amount of false positives and false negatives are not same).

$$\boldsymbol{f1 - score} = \frac{2 * (\mathrm{Recall} * \mathrm{Precision})}{(\mathrm{Recall} + \mathrm{Precision})} \tag{2.20}$$

Area Under the ROC Curve (AUC):

Area Under the ROC Curve (AOC) is a crucial tool for diagnosing test assessment. and is a plot between the true positive rate or sensitivity $\left(\frac{\mathbf{TP}}{\mathbf{TP+FN}}\right)$ and the false positive rate $\left(\frac{\mathbf{FP}}{\mathbf{FP+TN}}\right)$ for the various diagnostic test cut-off values that could be used. It is used in the cases of binary class classification. AUC score ranges between 0 and 1. An AUC score of 0.0 indicates a model with 100% erroneous predictions, whereas an AUC score of 1.0 indicates a model with 100% correct predictions.

2.5.1 WAVELET TRANSFORM + VGG16 BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of ML classifiers based on Wavelet Transform and VGG16 based extracted features.

Table I. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.89	0.86	0.90	0.88
SVM	0.93	0.95	0.90	0.92
Logistic Regression	0.89	0.86	0.90	0.88
KNN	0.93	0.90	0.95	0.93
Soft Voting Classifier	**0.96**	**0.95**	**0.95**	**0.95**

Table I. Performance evaluation of different classifiers based on Wavelet Transform+VGG16 based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 8 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table I, it has been shown that Soft Voting Classifier has provided the maximum accuracy of 95.65%. It has also provided good precision, recall and f1-score with a value of 0.95. Before applying majority voting ensemble, SVM and KNN have provided the maximum accuracy of 93.47%. But with KNN algorithm False Negative (FN) is 1, where with SVM algorithm FN is 2. From figure 2.19 it has been shown that KNN's AUC score (0.937) is better than SVM's AUC score (0.931). So KNN has provided better classification result than SVM. It has been noticed that majority voting ensemble improved the evaluation metrics of the classification.

Figure 2.17 shows the confusion matrix of ML classifiers.

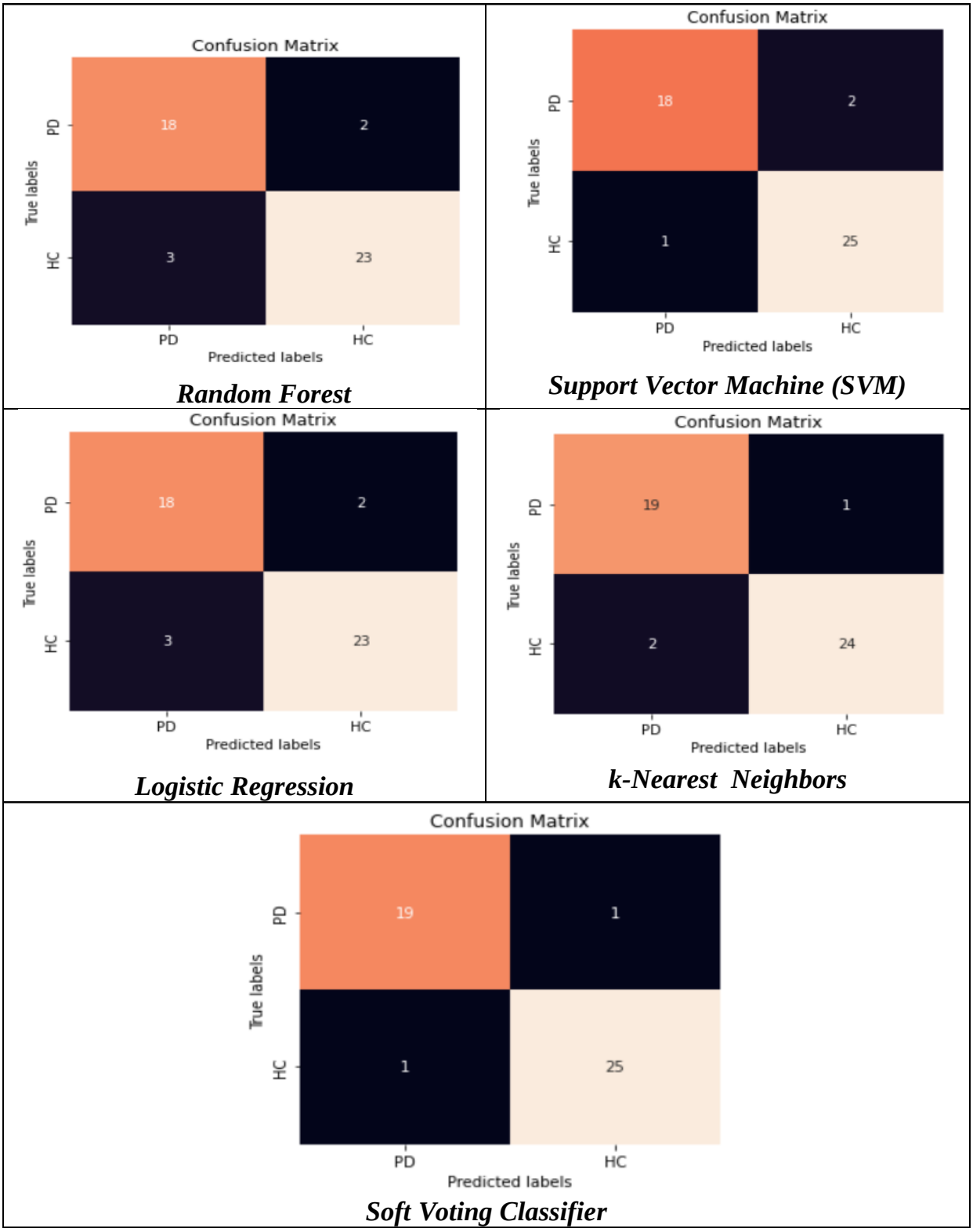


Fig 2.17. Confusion matrix using different Machine Learning classifiers based on Wavelet Transform and VGG16 based extracted features

Figure 2.18 represents the AUC graph and AUC score of ML classifiers. From the figure it has been shown that Soft Voting classifier has provided the maximum AUC score of 0.956.

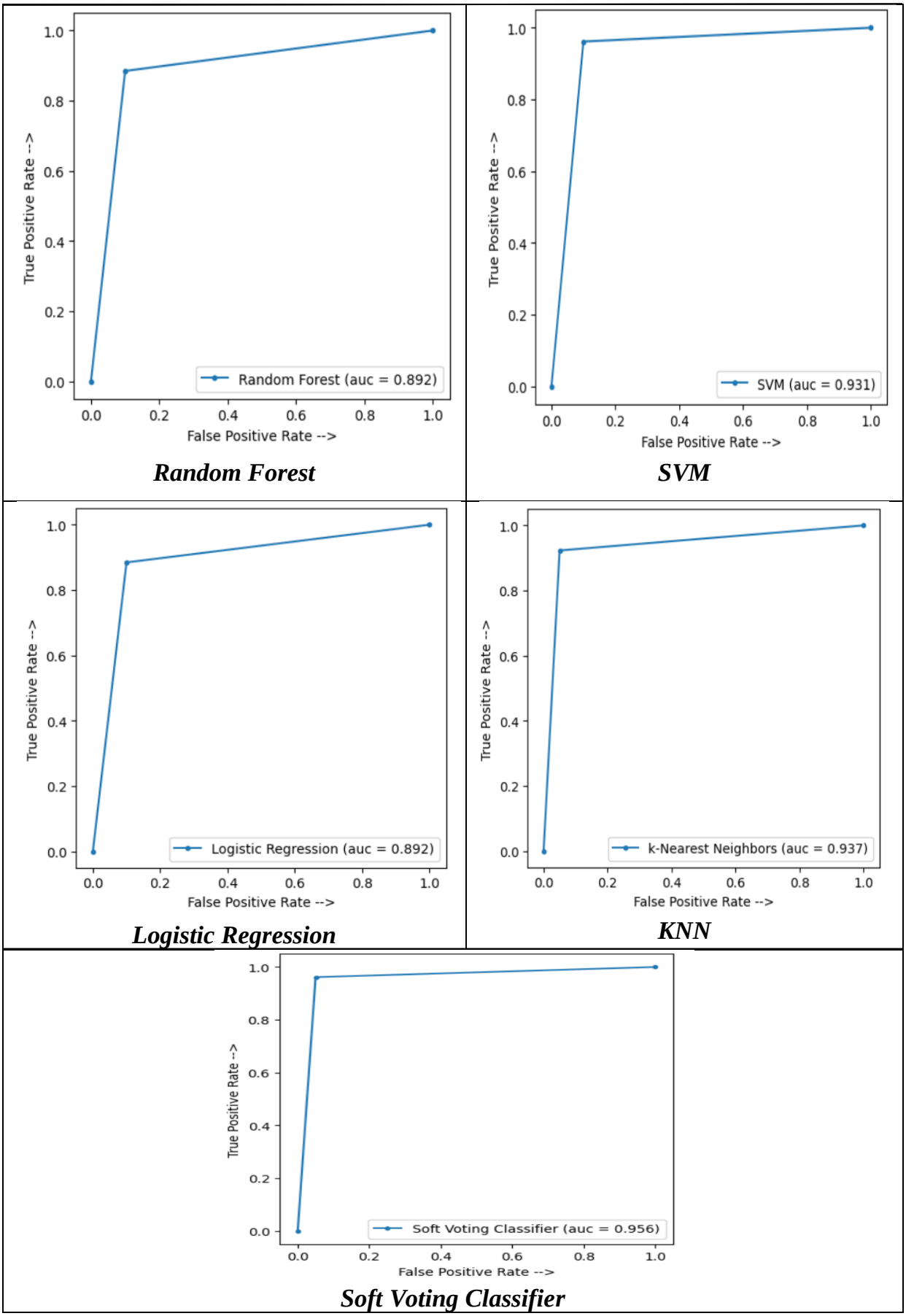


Fig 2.18. AUC graphs of different ML classifiers

2.5.2 WAVELET TRANSFORM + VGG19 BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Wavelet Transform and VGG19 based extracted features.

Table II. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.89	0.86	0.90	0.88
SVM	0.89	0.83	0.95	0.88
Logistic Regression	0.85	0.78	0.90	0.84
KNN	0.87	0.89	0.80	0.84
Soft Voting Classifier	**0.91**	**0.86**	**0.95**	**0.90**

Table II. Performance evaluation of different ML classifiers based on Wavelet Transform and VGG19 based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 8 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table II, it has been shown that Soft Voting Classifier has provided the maximum accuracy of 91.30%. It has also provided good precision, recall and f1-score of 0.86, 0.95 and 0.90 respectively. Before applying majority voting ensemble, Random Forest and SVM have provided the maximum accuracy of 89.13%. But SVM has provided better classification result than Random Forest with less FN (1) and better AUC score (0.898). It has been noticed that majority voting ensemble improved the evaluation metrics of the classification.

Figure 2.19 shows the confusion matrix of all ML classifiers.

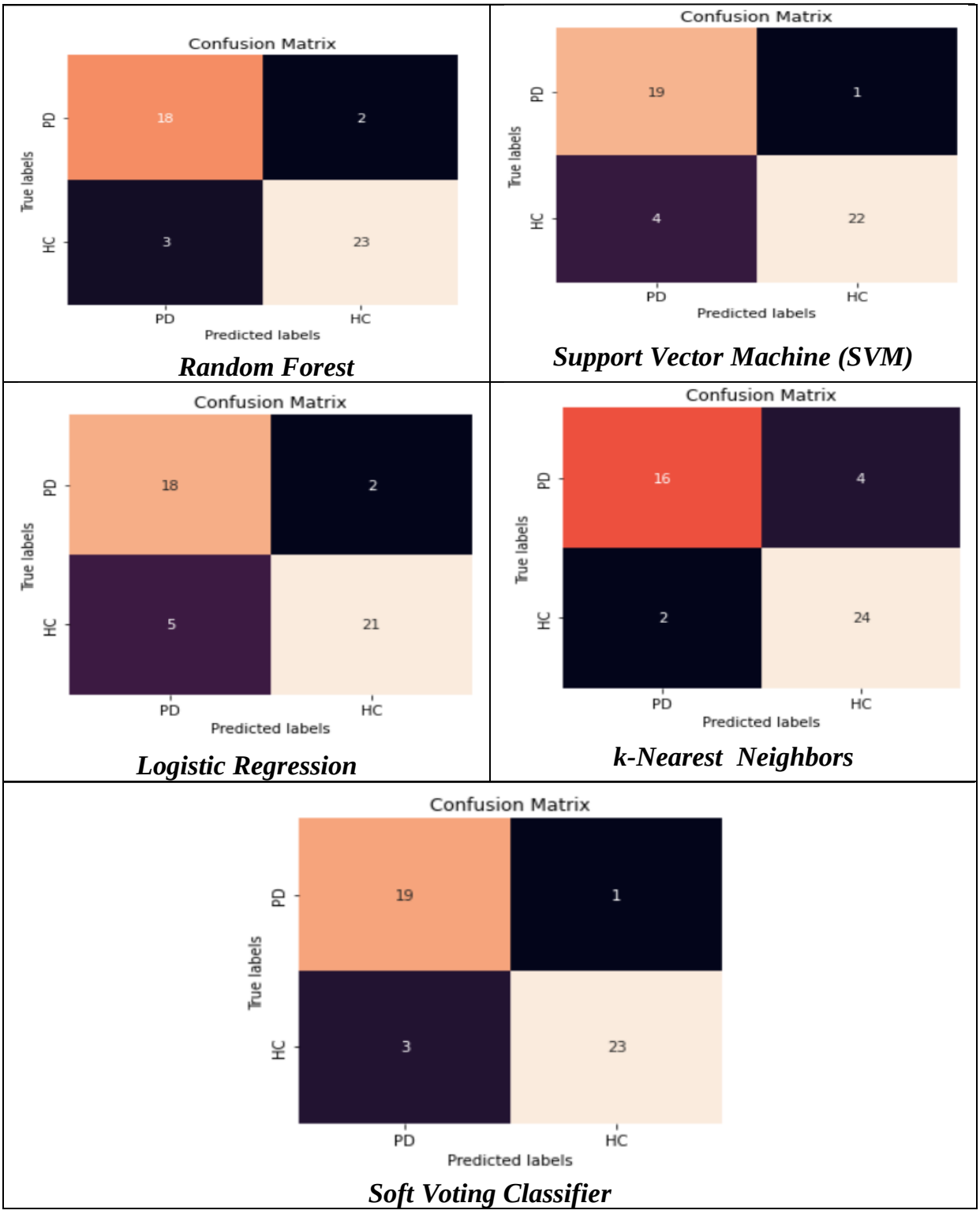


Fig 2.19. Confusion matrix using different Machine Learning classifiers based on Wavelet Transform and VGG19 based extracted features

Figure 2.20 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Soft Voting classifier has provided the maximum AUC score of 0.917.

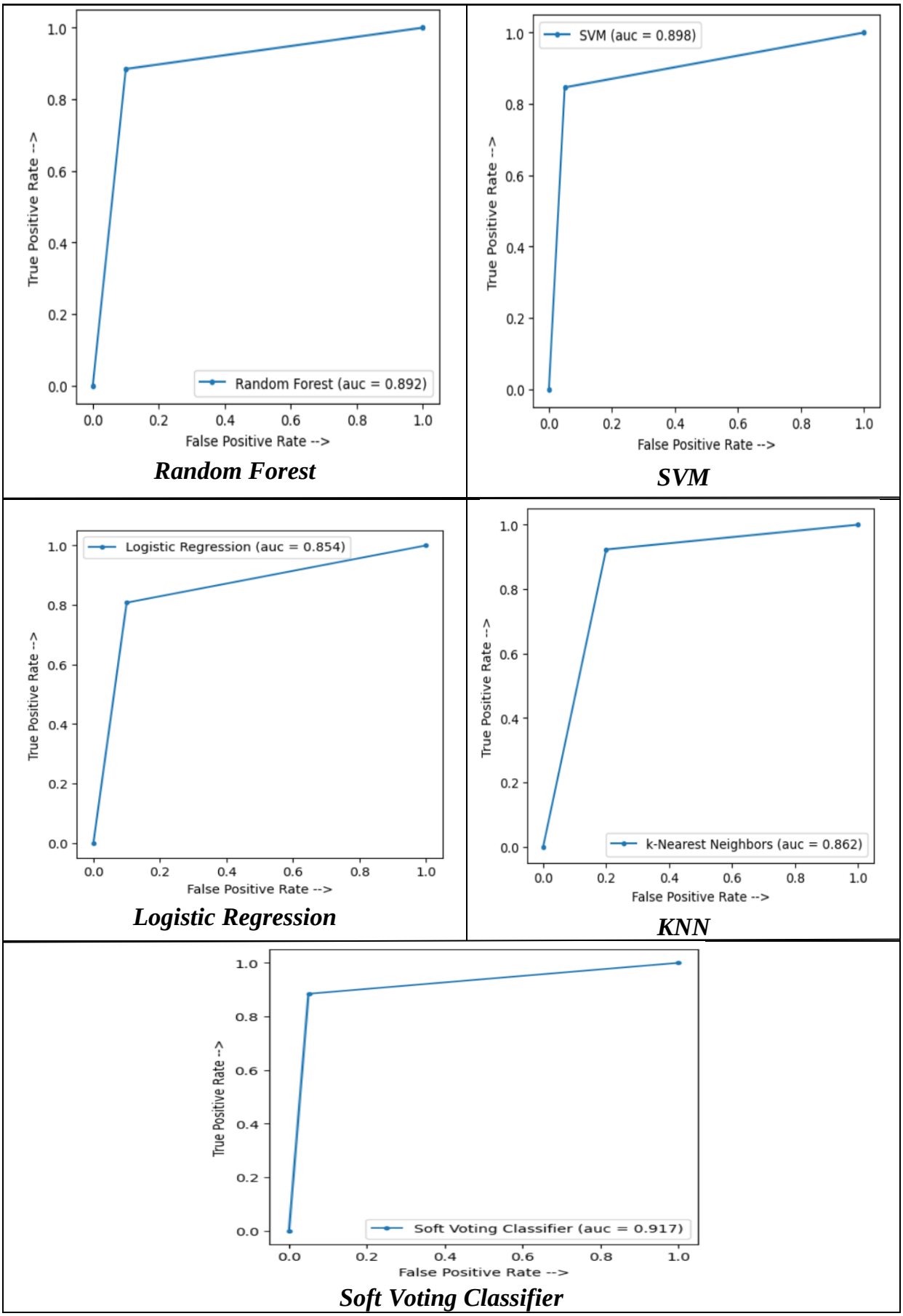


Fig 2.20. AUC graphs of different ML classifiers

2.5.3 WAVELET TRANSFORM+ DenseNet-121 BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Wavelet Transform and DensNet-121 based extracted features.

Table II. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.87	0.85	0.85	0.85
SVM	0.87	0.89	0.80	0.84
Logistic Regression	0.83	0.80	0.80	0.80
KNN	0.83	0.83	0.75	0.79
Soft Voting Classifier	**0.89**	**0.94**	**0.80**	**0.86**

Table III. Performance evaluation of different ML classifiers based on Wavelet Transform and DenseNet-121 based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 10 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table III, it has been shown that Soft Voting Classifier has provided the maximum accuracy of 89.13%. It has provided precision, recall and f1-score with a value of 0.94, 0.80 and 0.86 respectively. Before applying majority voting ensemble, Random Forest and SVM has provided the maximum accuracy of 87%. But Random Forest has provided better classification result than SVM with less FN (3) and better AUC score (0.867). It has been noticed that majority voting ensemble improved the accuracy of classification.

Figure 2.21 shows the confusion matrix of all ML classifiers.

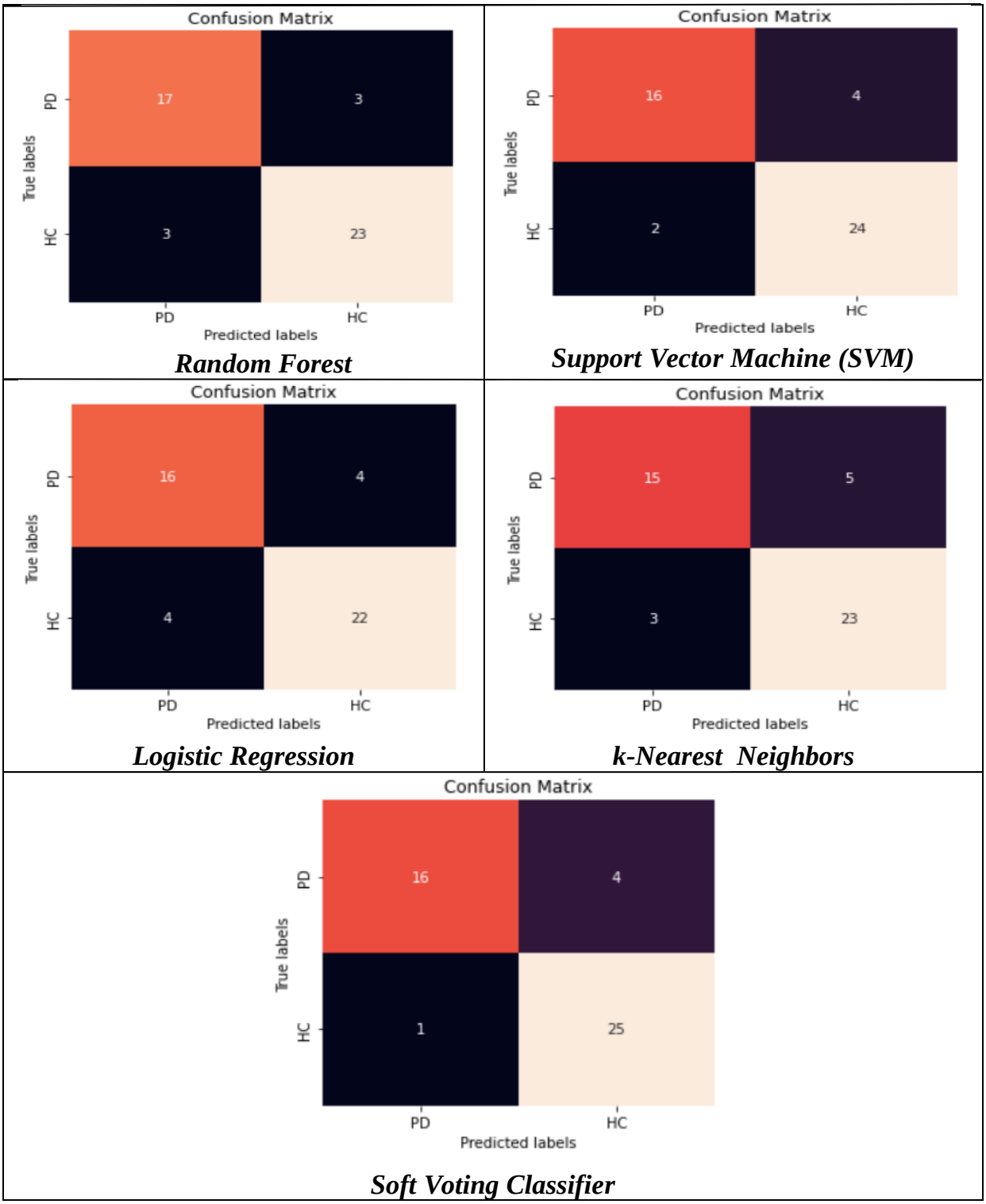


Fig 2.21. Confusion matrix using different Machine Learning classifiers based on Wavelet Transform and DenseNet-121 based extracted features

Figure 2.22 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Soft Voting classifier has provided the maximum AUC score of 0.881.

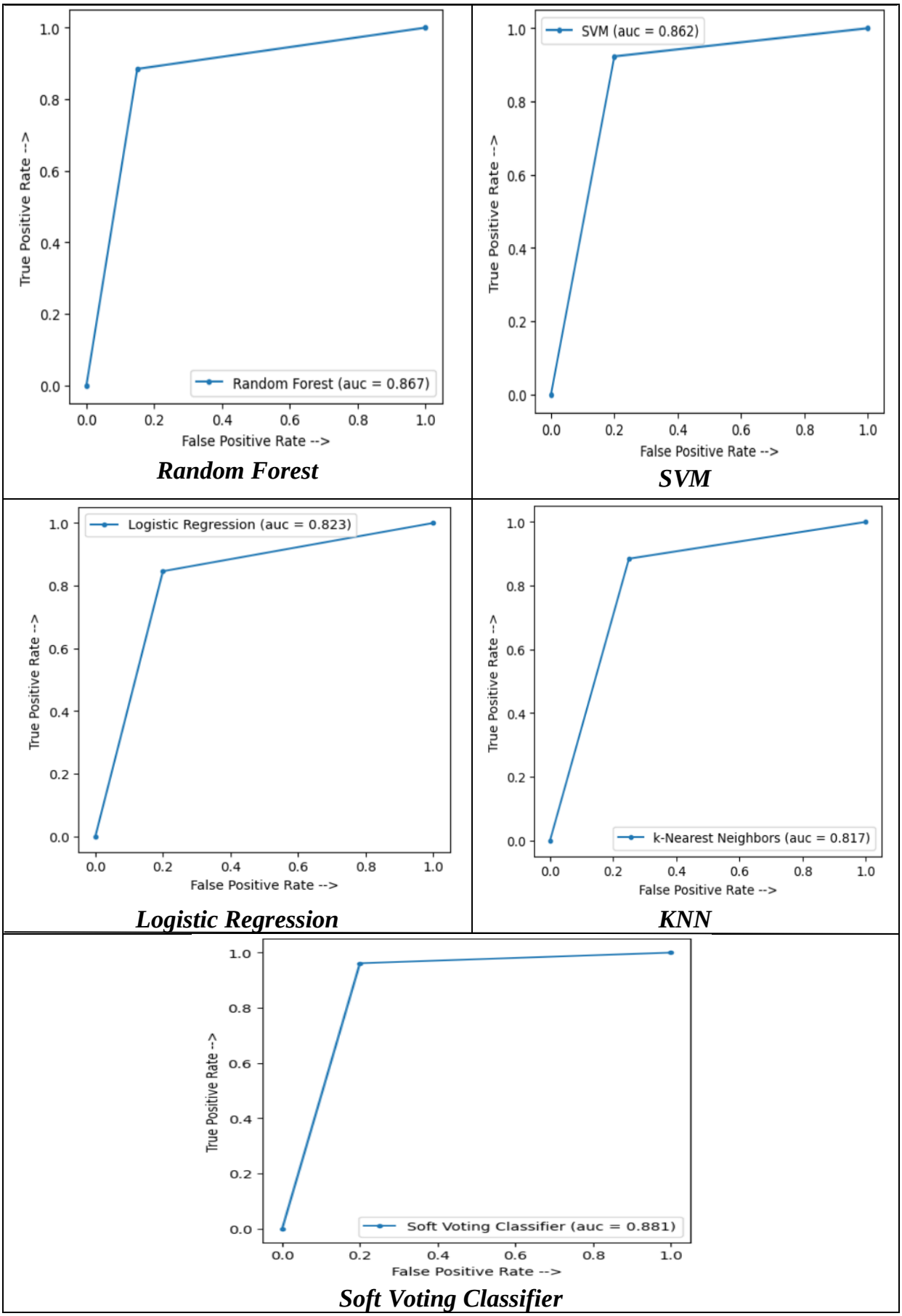


Fig 2.22. AUC graphs of different ML classifiers

2.5.4 WAVELET TRANSFORM+ XCEPTION BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Wavelet Transform and Xception based extracted features.

Table IV. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.93	0.95	0.90	0.92
SVM	0.91	0.90	0.90	0.90
Logistic Regression	0.85	0.84	0.80	0.82
KNN	0.91	1	0.80	0.89
Soft Voting Classifier	**0.89**	**0.89**	**0.85**	**0.87**

Table IV. Performance evaluation of different ML classifiers based on Wavelet Transform and Xception based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 8 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table IV, it has been shown that Random Forest has provided the maximum accuracy of 93.47%. It has also provided good precision, recall and f1-score with a value of 0.95, 0.90 and 0.92 respectively.

Figure 2.23 shows the confusion matrix of all ML classifiers.

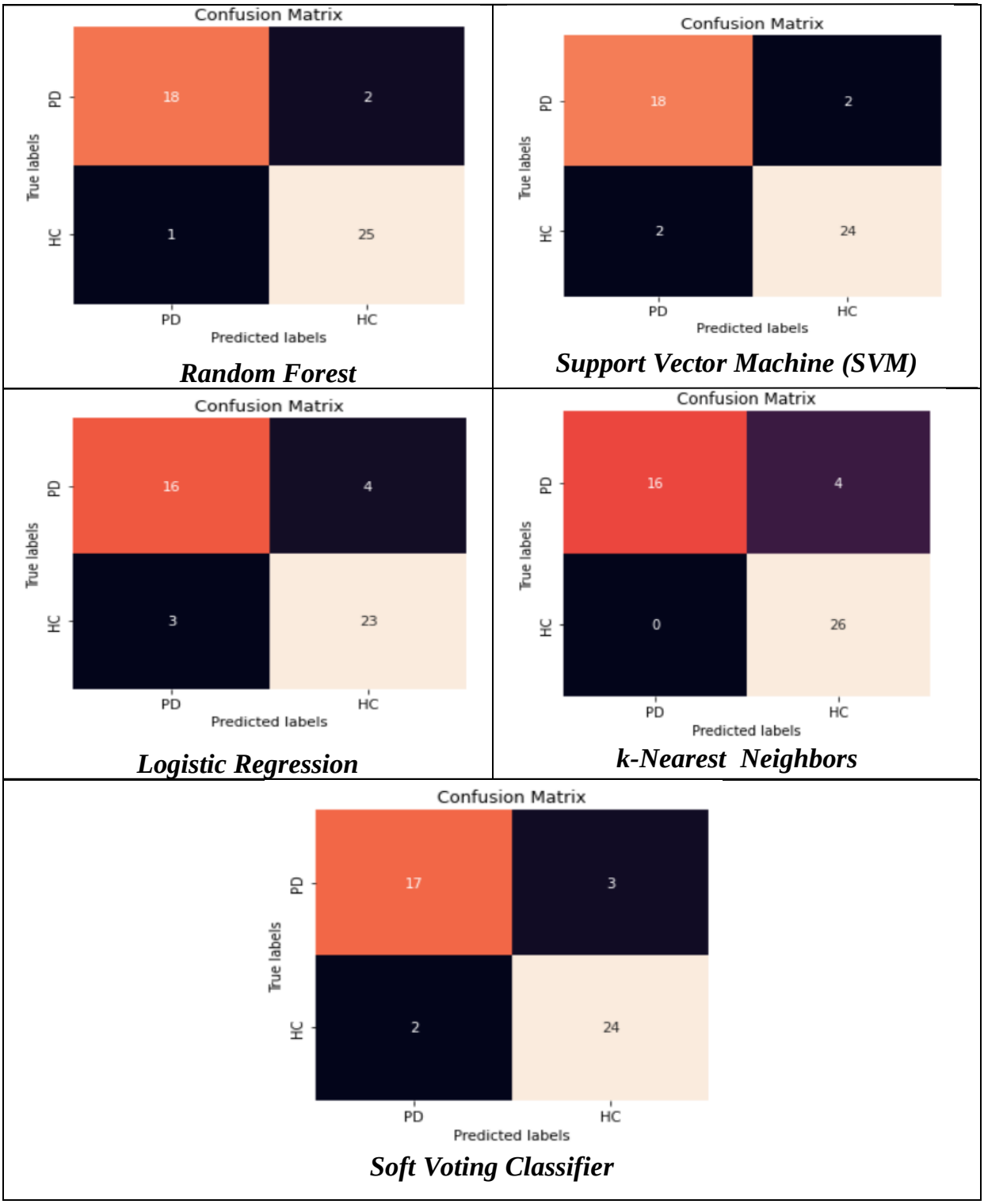


Fig 2.23. Confusion matrix using different Machine Learning classifiers based on Wavelet Transform and Xception based extracted features

Figure 2.24 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Random Forest classifier has provided the maximum AUC score of 0.932.

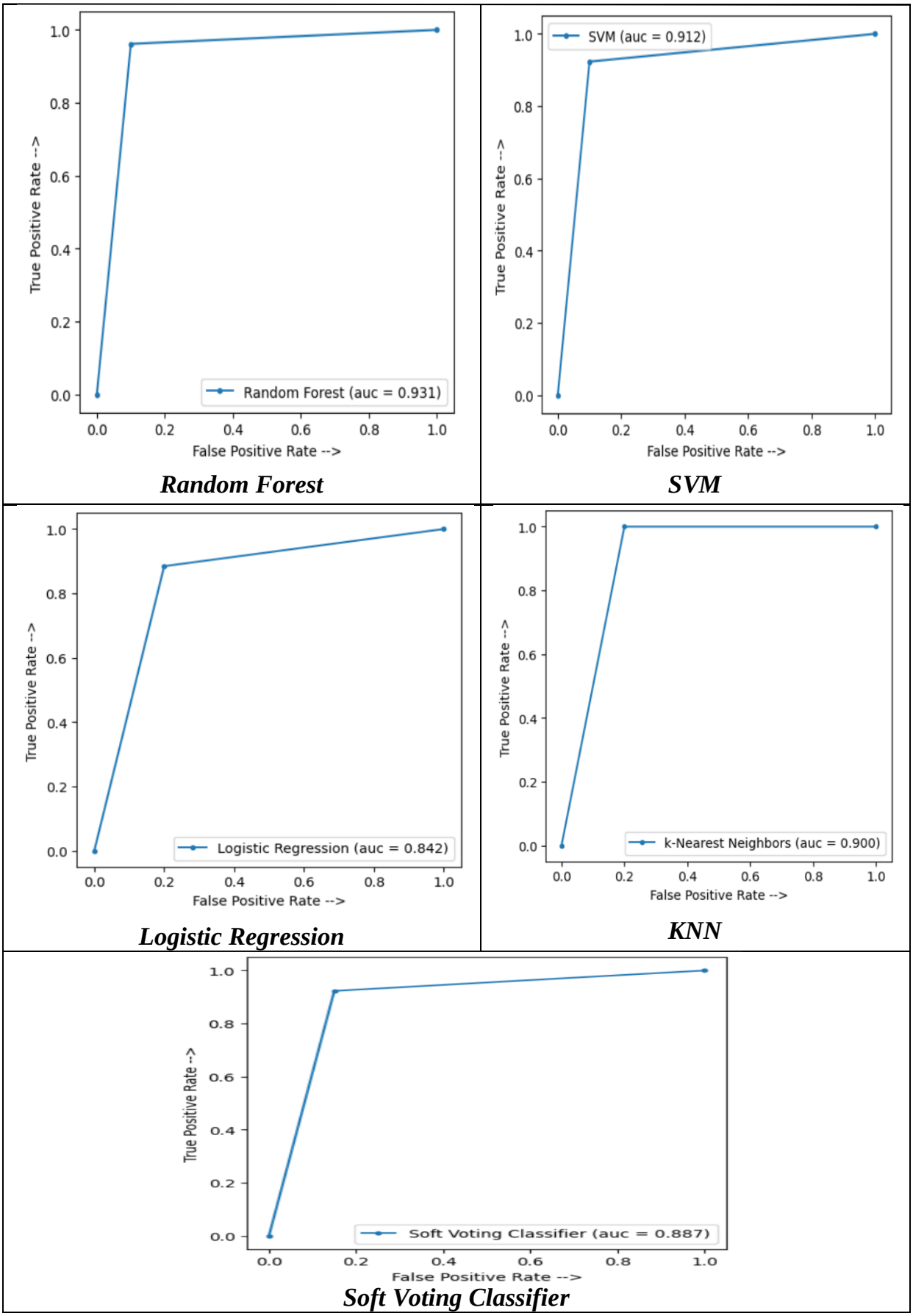


Fig 2.24. AUC graphs of different ML classifiers

2.5.5 CROSS-WAVELET TRANSFORM+ VGG16 BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Cross-wavelet Transform and VGG16 based extracted features.

Table V. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.89	0.94	0.80	0.86
SVM	0.80	0.74	0.85	0.79
Logistic Regression	0.76	0.75	0.71	0.73
KNN	0.76	0.80	0.60	0.69
Soft Voting Classifier	**0.78**	**0.75**	**0.75**	**0.75**

Table V. Performance evaluation of different ML classifiers based on Cross-wavelet Transform and VGG16 based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 3 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table V, it has been shown that Random Forest has provided the maximum accuracy of 89.13%. It has also provided good precision, recall and f1-score with a value of 0.94, 0.80 and 0.86 respectively.

Figure 2.25 shows the confusion matrix of all ML classifiers.

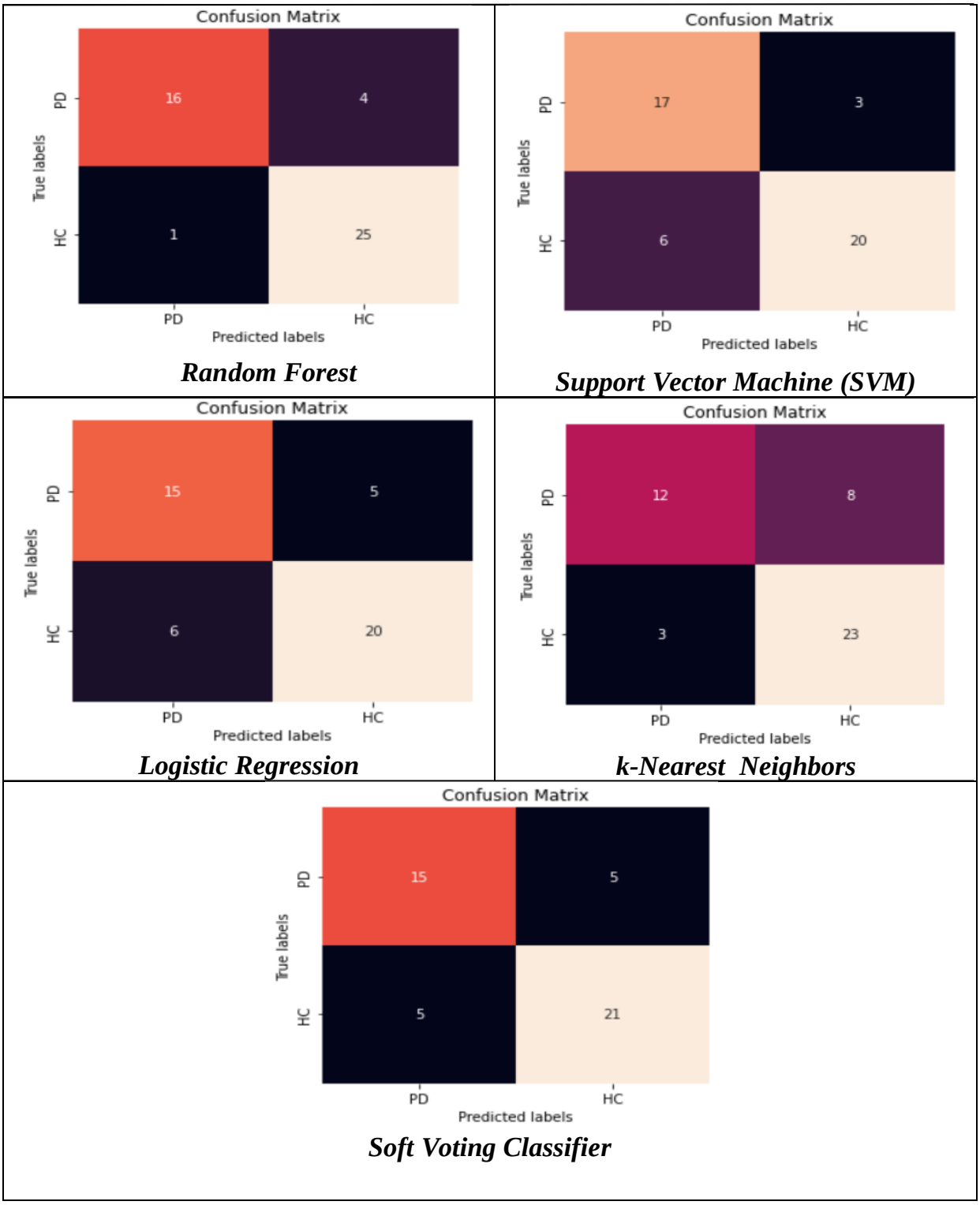


Fig 2.25. Confusion matrix using different Machine Learning classifiers based on Cross-wavelet Transform and VGG16 based extracted features

Figure 2.26 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Random Forest classifier has provided the maximum AUC score of 0.881.

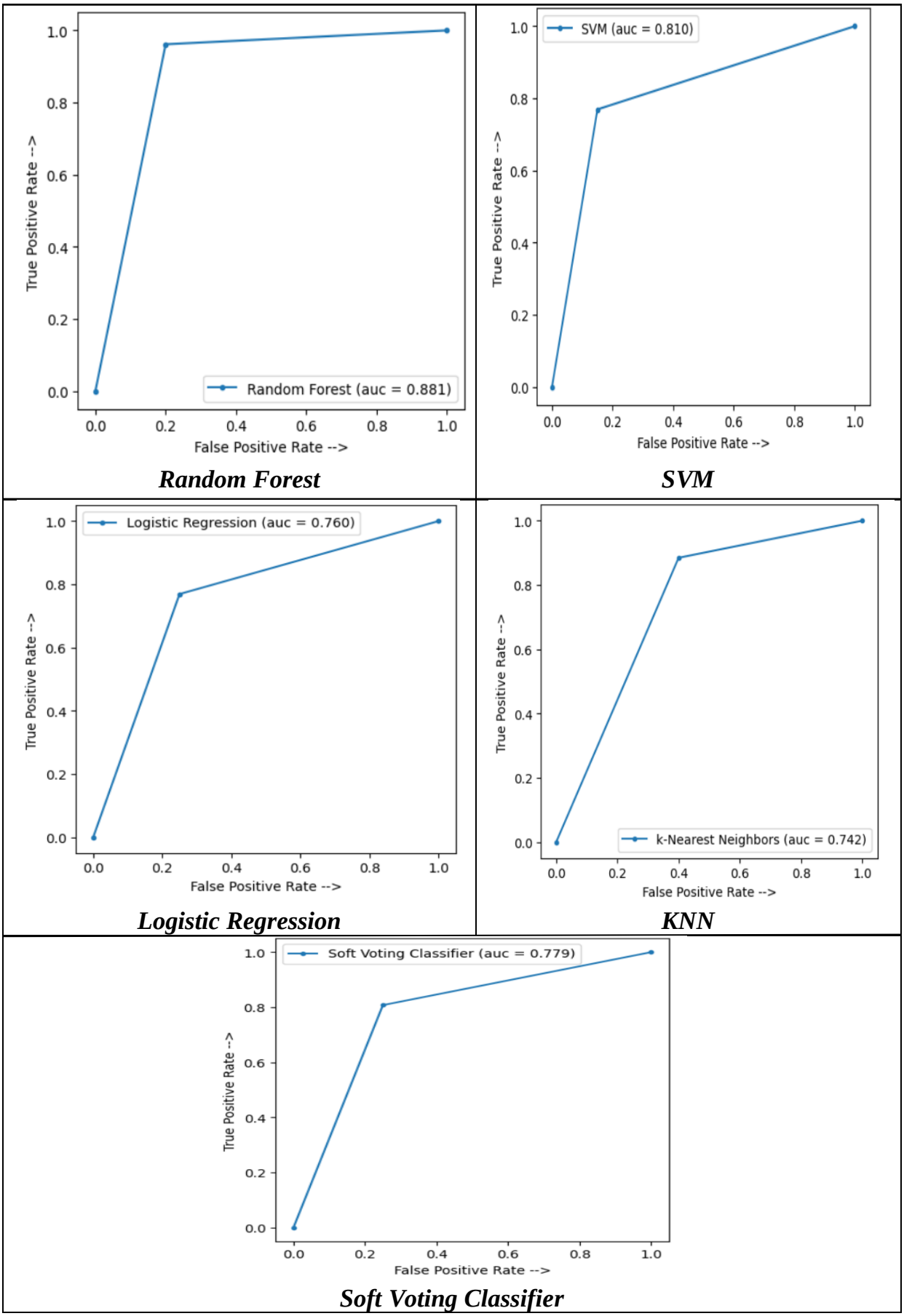


Fig 2.26. AUC graphs of different ML classifiers

2.5.6 CROSS-WAVELET TRANSFORM+ VGG19 BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Cross-wavelet Transform and VGG19 based extracted features.

Table VI. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.85	0.84	0.80	0.82
SVM	0.83	0.80	0.80	0.80
Logistic Regression	0.83	0.77	0.85	0.81
KNN	0.72	0.68	0.65	0.67
Soft Voting Classifier	**0.83**	**0.77**	**0.85**	**0.81**

Table VI. Performance evaluation of different ML classifiers based on Cross-wavelet Transform and VGG19 based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 2 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table VI, it has been shown that Random Forest has provided the maximum accuracy of 84.78%. It has also provided precision, recall and f1-score of 0.84, 0.80 and 0.82 respectively.

Figure 2.27 shows the confusion matrix of all ML classifiers.

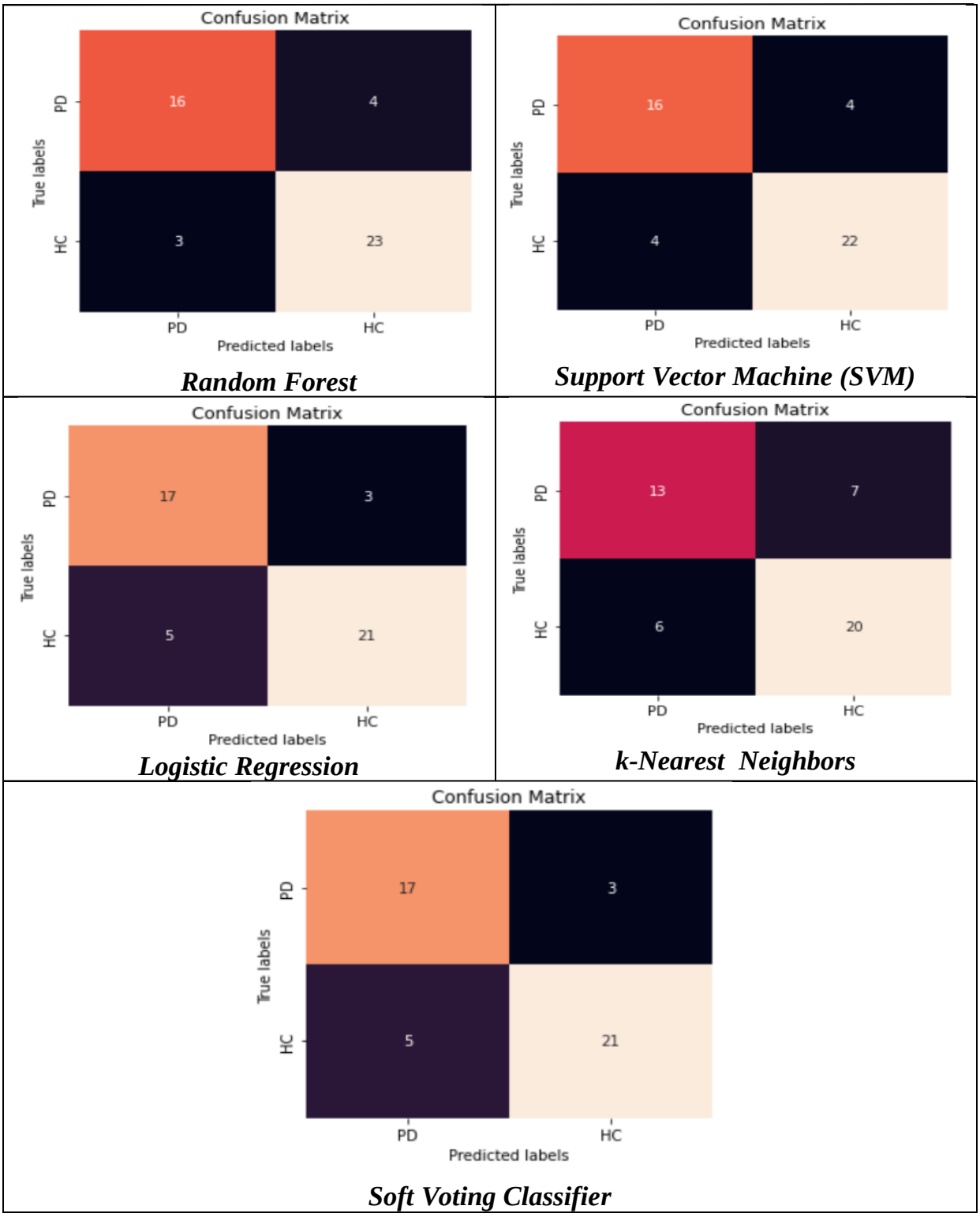


Fig 2.27. Confusion matrix using different Machine Learning classifiers based on Cross-wavelet Transform and VGG19 based extracted features

Figure 2.28 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Random Forest classifier has provided the maximum AUC score of 0.842.

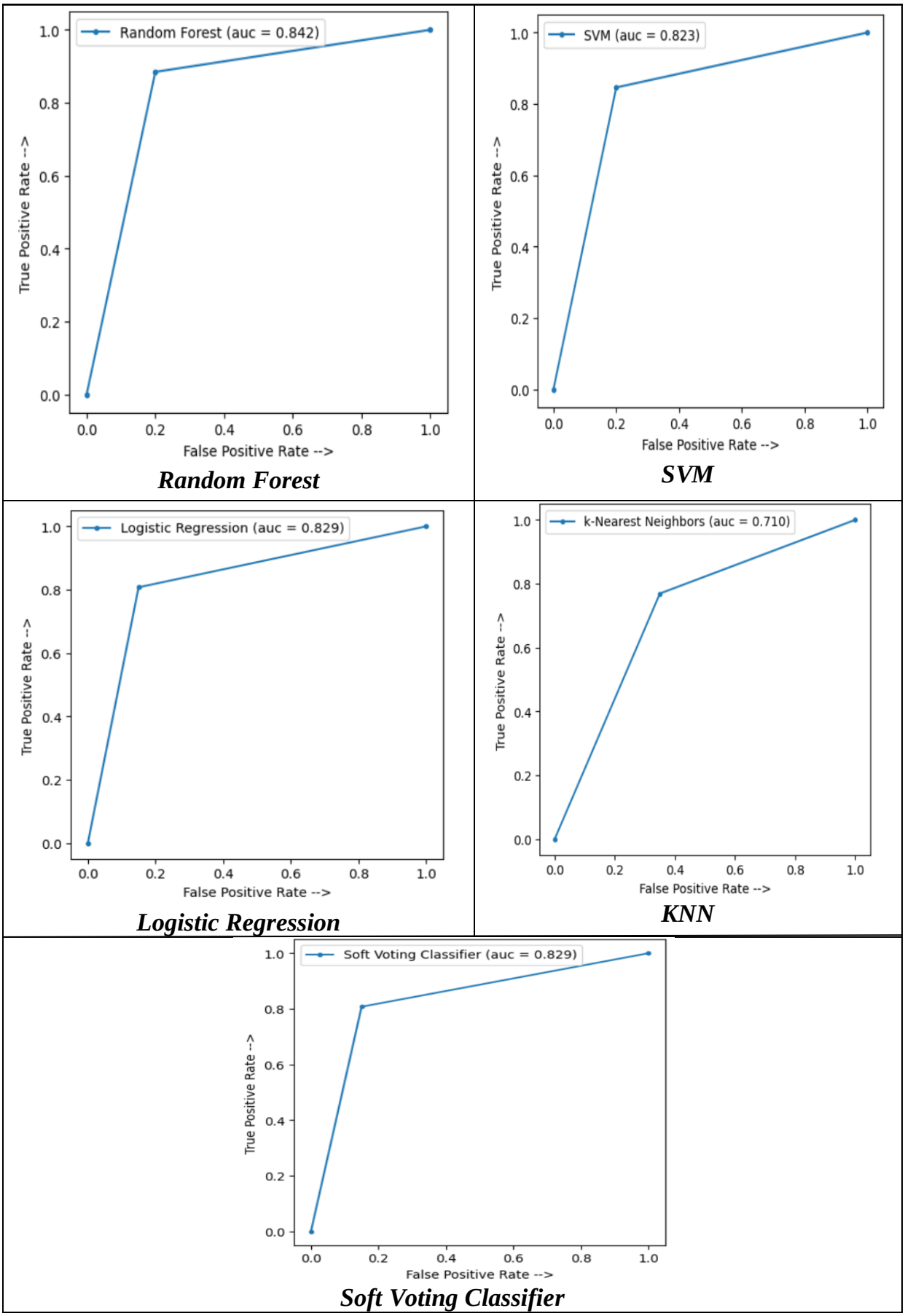


Fig 2.28. AUC graphs of different ML classifiers

2.5.7 CROSS-WAVELET TRANSFORM+ DenseNet-121 BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Cross-wavelet Transform and DenseNet-121 based extracted features.

Table VII. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.91	0.90	0.90	0.90
SVM	0.87	0.85	0.85	0.85
Logistic Regression	0.83	0.80	0.80	0.80
KNN	0.85	0.84	0.80	0.82
Soft Voting Classifier	**0.89**	**0.89**	**0.85**	**0.87**

Table VII. Performance evaluation of different ML classifiers based on Cross-wavelet Transform and DenseNt-121 based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 9 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table VII, it has been shown that Random Forest has provided the maximum accuracy of 91.30% with precision, recall and f1-score of 0.90.

Figure 2.29 shows the confusion matrix of all ML classifiers.

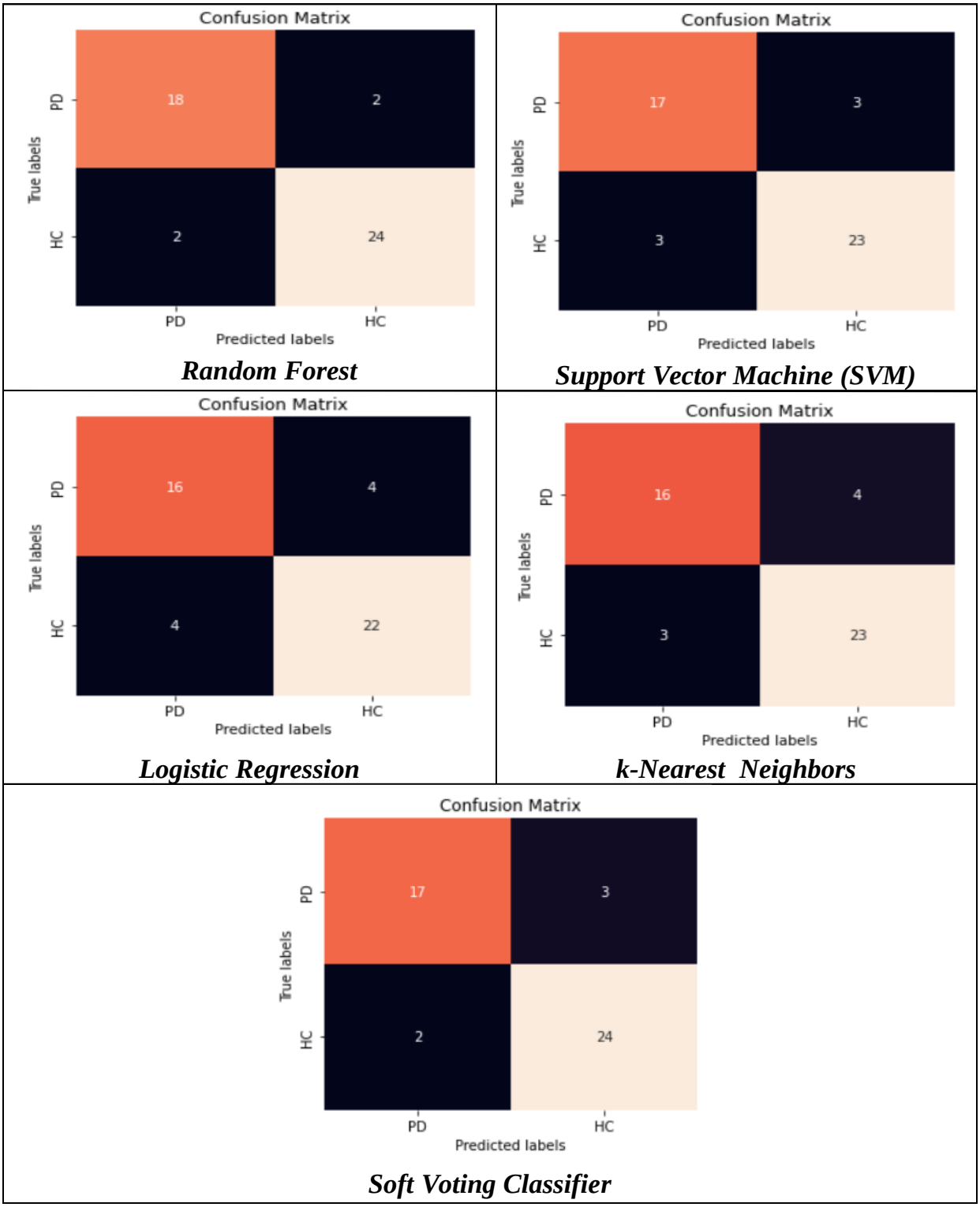


Fig 2.29. Confusion matrix using different Machine Learning classifiers based on Cross-wavelet Transform and DenseNet-121 based extracted features

Figure 2.30 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Random Forest classifier has provided the maximum AUC score of 0.912.

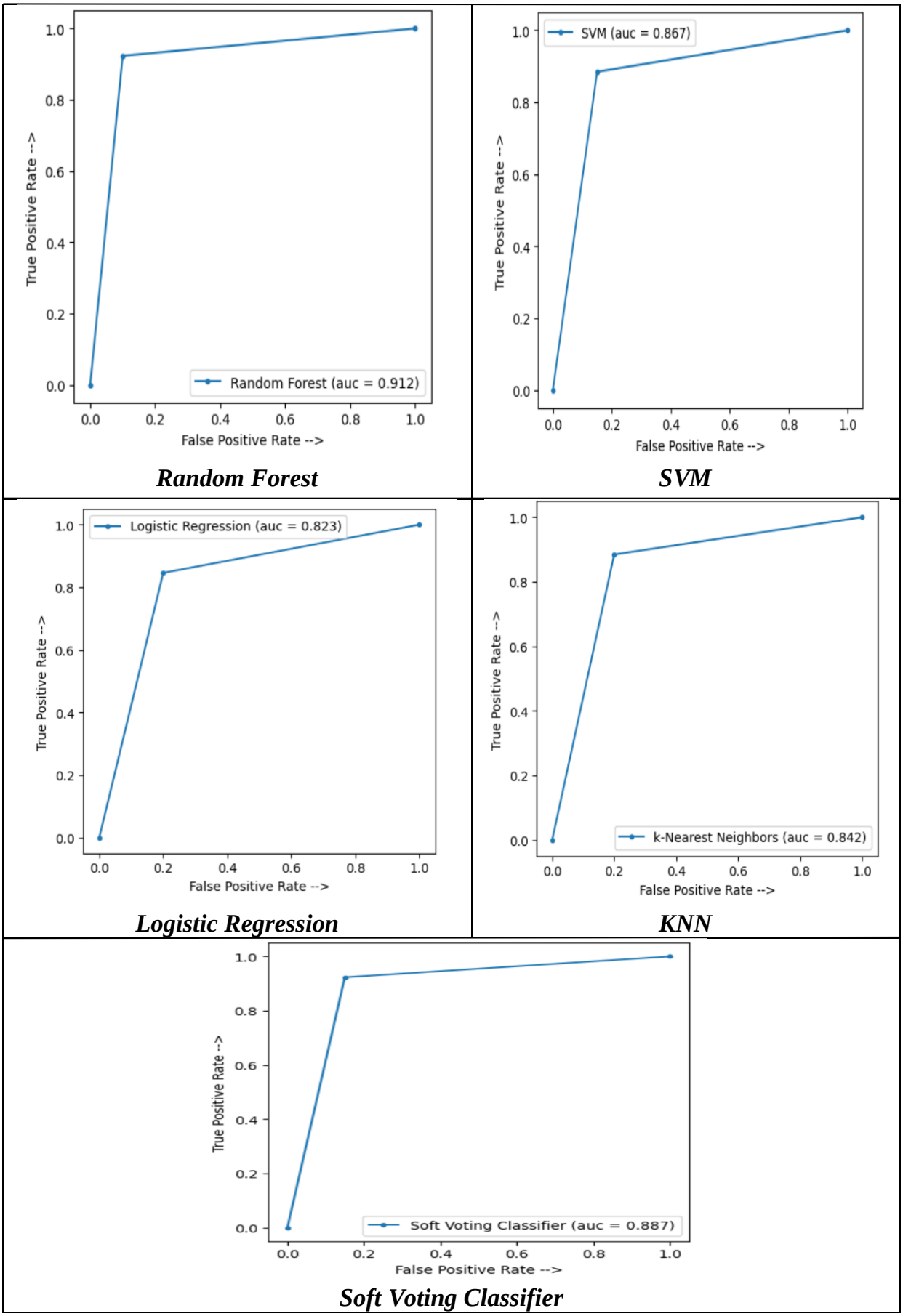


Fig 2.30. AUC graphs of different ML classifiers

2.5.8 CROSS-WAVELET TRANSFORM+ XCEPTION BASED RESULT

This section shows the evaluation metrics, confusion matrix and AUC graphs of different ML classifiers based on Cross-wavelet Transform and Xception based extracted features.

Table VIII. represents the evaluation metrics of different ML classifiers which were used for PD classification problem.

Algorithms	Accuracy	Precision	Recall	f1-score
Random Forest	0.85	0.78	0.90	0.84
SVM	0.89	0.86	0.90	0.88
Logistic Regression	0.93	0.87	1.00	0.93
KNN	0.85	0.88	0.75	0.81
Soft Voting Classifier	**0.89**	**0.87**	**1.00**	**0.93**

Table VIII. Performance evaluation of different ML classifiers based on Cross-wavelet Transform and Xception based extracted features

For SVM, hyperparameter C has been chosen as 13 and 'rbf' kernel has been used. The number of trees (n_estimators) was chosen as 200 in the Random Forest. For KNN, 8 nearest neighbours (n_neighbors) have been chosen. Soft Voting classifier has been used as majority voting ensembling.

From Table VIII, it has been shown that Logistic Regression and Soft Voting classifier have provided the maximum accuracy of 93.47% with precision, recall and f1-score of 0.87, 1 and 0.93 respectively. They have also provided 0 FN case.

Figure 2.31 shows the confusion matrix of all ML classifiers.

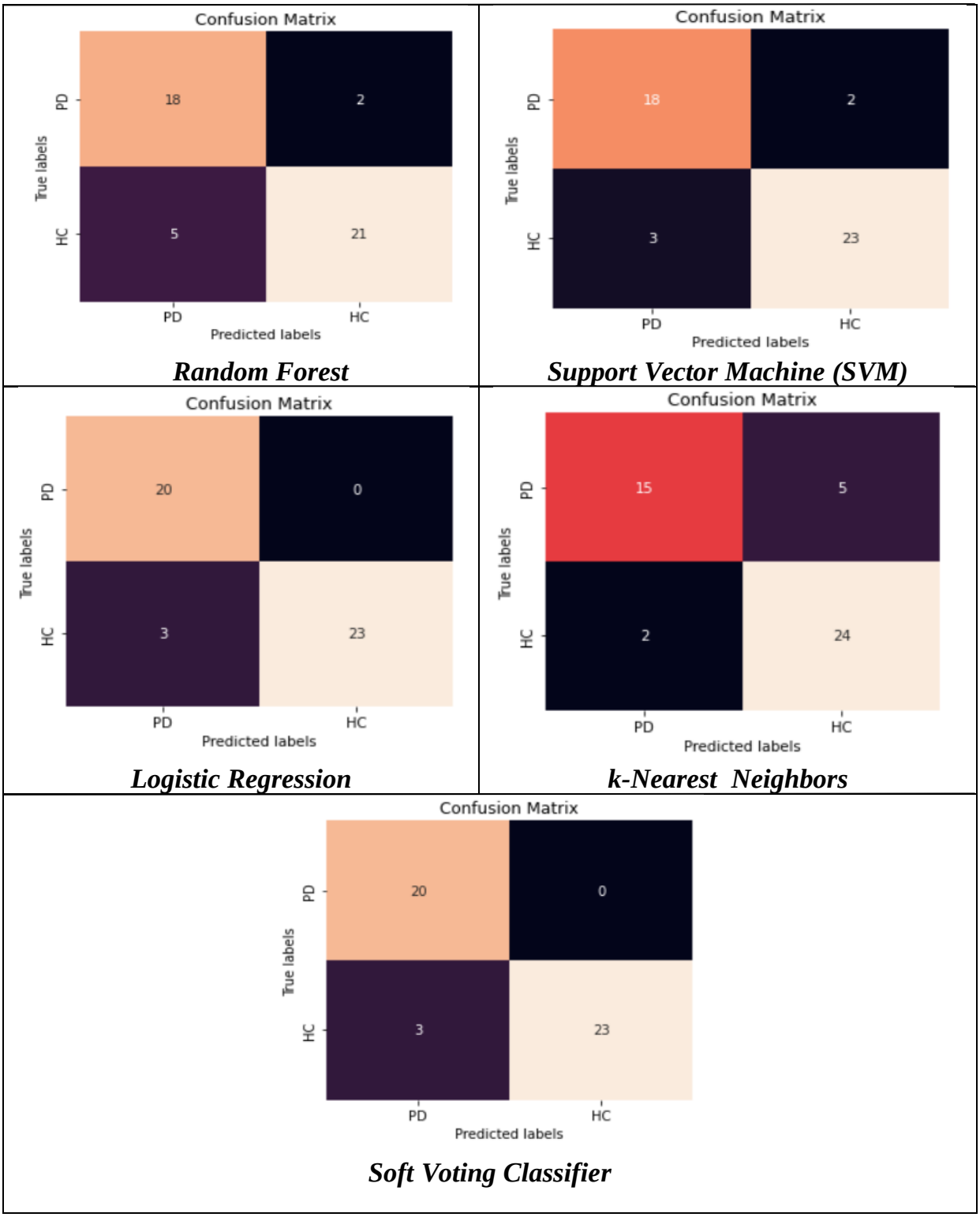


Fig 2.31. Confusion matrix using different Machine Learning classifiers based on Cross-wavelet Transform and DenseNet-121 based extracted features

Figure 2.32 represents the AUC graph and AUC score of different ML classifiers. From the figure it has been shown that Logistic Regression and Soft Voting classifier have provided the maximum AUC score of 0.912.

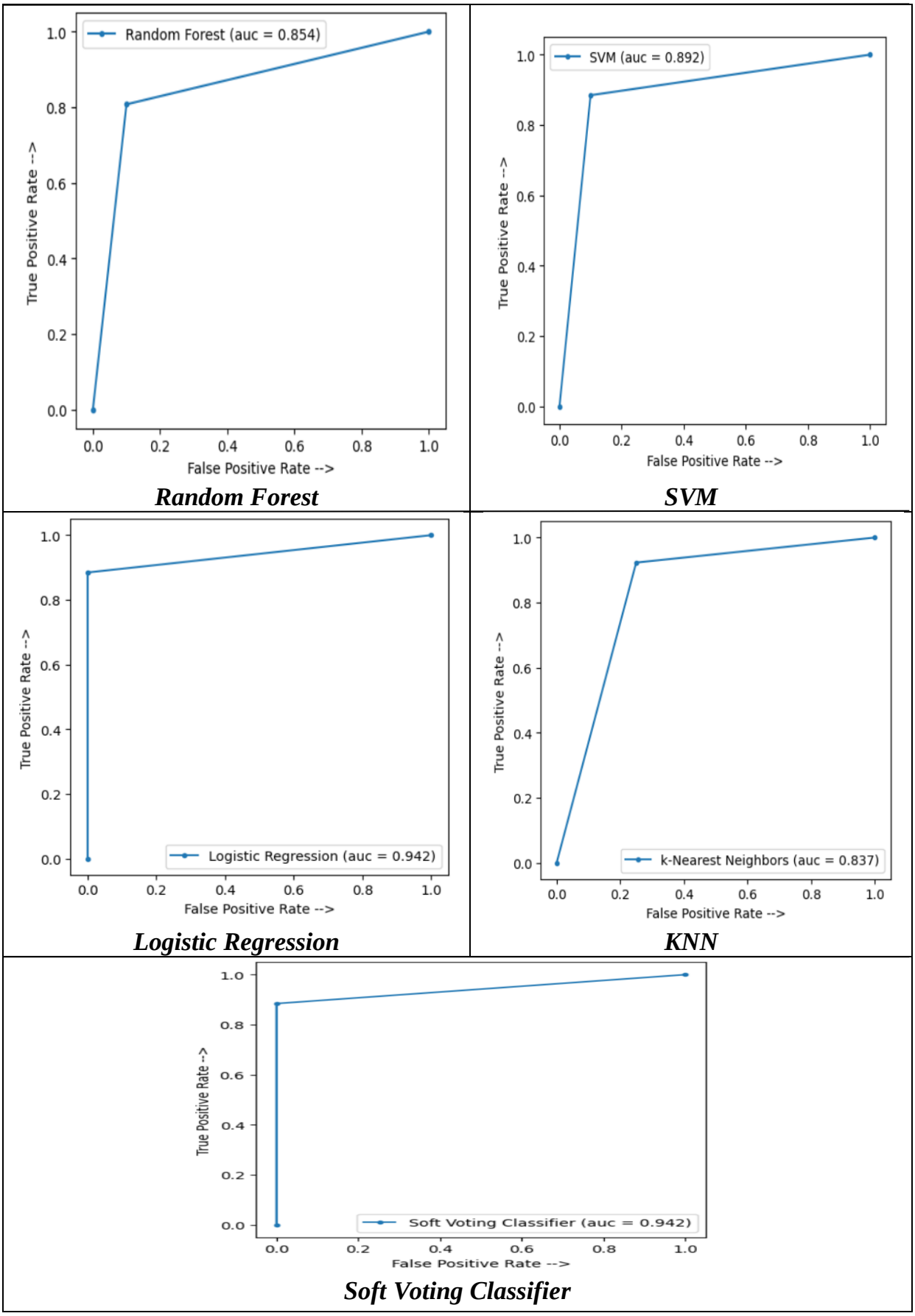


Fig 2.32. AUC graphs of different ML classifiers

2.6 CONCLUSIONS

In this chapter, audio signals have been used for Parkinson's disease (PD) classification. This chapter represented a feature extraction technique which involved Wavelet Transform and Cross-wavelet Transform with VGG16, VGG19, DenseNet-121 and Xception architecture. Then multiple Machine Learning algorithms were implemented for classification problem.

Wavelet Transform with VGG16 based feature extraction technique has provided the best evaluation metrics (accuracy, precision, recall, f1-score and AUC score). In this feature extraction method, Soft Voting classifier has provided the maximum accuracy of 95.65% with precision, recall and f1-score of 0.95. Other proposed feature extraction techniques have also provided satisfactory classification results. From figure 2.32, it has been noticed that Cross-wavelet transform with Xception based feature extraction method has produced 0 FN case with Logistic Regression and Soft Voting Classifier. In medical diagnostic problems, FN is more severe than FP. In this chapter, it has been noticed that Xception and DenseNet-121 architecture have extracted better features from the Cross-wavelet scalograms than VGG16 and VGG19. Due to this they have provided better evaluation metrics.

These proposed feature extraction techniques have provided better classification result than the traditionally used Mel-frequency Cepstral Coefficients (MFCC) based featue extraction technique.

References

[1] Tanner, Poewe W. Seppi K. "Halliday GM Brundin P Volkmann J Schrag AE Lang AE Parkinson disease." *Nat Rev Dis Primers* 3.17013 (2017): 10-1038

[2] Ackermann, Hermann, Steffen R. Hage, and Wolfram Ziegler. "Brain mechanisms of acoustic communication in humans and nonhuman primates: An evolutionary perspective." *Behavioral and Brain Sciences* 37.6 (2014): 529-546.

[3] Foppa, Aline Aparecida, et al. "Medication therapy management service for patients with Parkinson's disease: a before-and-after study." *Neurology and therapy* 5.1 (2016): 85-99.

[4] Khoshnevis, Seyed Alireza, and Ravi Sankar. "Classification of the stages of Parkinson's disease using novel higher-order statistical features of EEG signals." *Neural Computing and Applications* 33.13 (2021): 7615-7627.

[5] Sakar, C. Okan, et al. "A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform." *Applied Soft Computing* 74 (2019): 255-263.

[6] Sapir, Shimon. "Multiple factors are involved in the dysarthria associated with Parkinson's disease: a review with implications for clinical practice and research." *Journal of Speech, Language, and Hearing Research* 57.4 (2014): 1330-1343.

[7] Galaz, Zoltan, et al. "Prosodic analysis of neutral, stress-modified and rhymed speech in patients with Parkinson's disease." *Computer methods and programs in biomedicine* 127 (2016): 301-317.

[8] Pawlukowska, Wioletta, et al. "Articulation disorders and duration, severity and L-dopa dosage in idiopathic Parkinson's disease." *Neurologia i Neurochirurgia Polska* 49.5 (2015): 302-306.

[9] Lirani-Silva, Camila, Lúcia Figueiredo Mourão, and Lilian Teresa Bucken Gobbi. "Dysarthria and Quality of Life in neurologically healthy elderly and patients with Parkinson's disease." *CoDAS*. Vol. 27. Sociedade Brasileira de Fonoaudiologia, 2015.

[10] Blumin, Joel H., Dana E. Pcolinsky, and Joseph P. Atkins. "Laryngeal findings in advanced Parkinson's disease." *Annals of Otology, Rhinology & Laryngology* 113.4 (2004): 253-258.

[11] Martens, Heidi, et al. "Reception of communicative functions of prosody in hypokinetic dysarthria due to Parkinson's disease." *Journal of Parkinson's Disease* 6.1 (2016): 219-229.

[12] Sachin, S., et al. "Clinical speech impairment in Parkinson's disease, progressive supranuclear palsy, and multiple system atrophy." *Neurology India* 56.2 (2008): 122.

[13] Chenausky, Karen, Joel MacAuslan, and Richard Goldhor. "Acoustic analysis of PD speech." *Parkinson's Disease* 2011 (2011).

[14] Schlede, Nina, et al. "Clinical EEG in cognitively impaired patients with Parkinson's Disease." *Journal of the neurological sciences* 310.1-2 (2011): 75-78.

[15] Swann, Nicole, et al. "Deep brain stimulation of the subthalamic nucleus alters the cortical profile of response inhibition in the beta frequency band: a scalp EEG study in Parkinson's disease." *Journal of Neuroscience* 31.15 (2011): 5721-5729.

[16] Klassen, B. T., et al. "Quantitative EEG as a predictive biomarker for Parkinson disease dementia." *Neurology* 77.2 (2011): 118-124.

[17] Swann, Nicole, et al. "Deep brain stimulation of the subthalamic nucleus alters the cortical profile of response inhibition in the beta frequency band: a scalp EEG study in Parkinson's disease." *Journal of Neuroscience* 31.15 (2011): 5721-5729.

[18] Yuvaraj, R., et al. "Detection of emotions in Parkinson's disease using higher order spectral features from brain's electrical activity." *Biomedical Signal Processing and Control* 14 (2014): 108-116.

[19] Yuvaraj, Rajamanickam, U. Rajendra Acharya, and Yuki Hagiwara. "A novel Parkinson's Disease Diagnosis Index using higher-order spectra features in EEG signals." *Neural Computing and Applications* 30.4 (2018): 1225-1235.

[20] Liu, Guotao, et al. "Complexity analysis of electroencephalogram dynamics in patients with Parkinson's disease." *Parkinson's Disease* 2017 (2017).

[21] Sakhavi, Siavash, Cuntai Guan, and Shuicheng Yan. "Learning temporal information for brain-computer interface using convolutional neural networks." *IEEE transactions on neural networks and learning systems* 29.11 (2018): 5619-5629.

[22] Oh, Shu Lih, et al. "A deep learning approach for Parkinson's disease diagnosis from EEG signals." *Neural Computing and Applications* 32.15 (2020): 10927-10933.

[23] Obukhov, Yu V., et al. "Electroencephalograms features of the early stage Parkinson's disease." *Pattern recognition and image analysis* 24.4 (2014): 593-604.

[24] Naghsh, Erfan, Mohamad Farzan Sabahi, and Soosan Beheshti. "Spatial analysis of EEG signals for Parkinson's disease stage detection." *Signal, Image and Video Processing* 14.2 (2020): 397-405.

[25] Arora, Siddharth, Ladan Baghai-Ravary, and Athanasios Tsanas. "Developing a large scale population screening tool for the assessment of Parkinson's disease using telephone-quality voice." *The Journal of the Acoustical Society of America* 145.5 (2019): 2871-2884.

[26] Sakar, Betul Erdogdu, et al. "Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings." *IEEE Journal of Biomedical and Health Informatics* 17.4 (2013): 828-834.

[27] Wodzinski, Marek, et al. "Deep learning approach to Parkinson's disease detection using voice recordings and convolutional neural network dedicated to image

classification." *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019.

[28] Orozco-Arroyave, Juan Rafael, et al. "New Spanish speech corpus database for the analysis of people suffering from Parkinson's disease." *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. 2014.

[29] Zahid, Laiba, et al. "A spectrogram-based deep feature assisted computer-aided diagnostic system for Parkinson's disease." *IEEE Access* 8 (2020): 35482-35495.

[30] Trinh, Nam, and O'Brien Darragh. "Pathological speech classification using a convolutional neural network." (2019).

[31] D. Trivedi H. Jaeger and M. Stadtschnitzer. 2019. Mobile Device Voice Recordings at King's College London (MDVR-KCL) from both early and advanced Parkinson's disease patients and healthy controls. https://doi.org/10.5281/zenodo.2867216.

[32] Konar, P., and P. Chattopadhyay. "Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs)." *Applied Soft Computing* 11.6 (2011): 4203-4211.

[33] Meynard, Adrien, and Bruno Torrésani. "Spectral analysis for nonstationary audio." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.12 (2018): 2371-2380.

[34] Morshuis, Peter HF. "Degradation of solid dielectrics due to internal partial discharge: some thoughts on progress made and where to go now." *IEEE Transactions on Dielectrics and Electrical Insulation* 12.5 (2005): 905-913.

[35] Russakovsky, O., et al. "Desafio de reconhecimento visual em grande escala do ImageNet." *International Journal of Computer Vision* 115.3 (2015): 211-252.

[36] Shams, Mahmoud, Amira Elsonbaty, and Wael ElSawy. "Arabic handwritten character recognition based on convolution neural networks and support vector machine." *arXiv preprint arXiv:2009.13450* (2020).

[37] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[38] Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[39] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[40] Sinha, V. Y. K. P. K., and V. Y. Kulkarni. "Efficient learning of random forest classifier using disjoint partitioning approach." *Proceedings of the World Congress on Engineering*. Vol. 2. 2013.

[41] Vapnik, Vladimir N. "The nature of statistical learning." *Theory* (1995).

CHAPTER 3

Bearing Fault Classification using Deep Learning Techniques

3.1 INTRODUCTION

In the recent time, industrial automation has accelerated the use of sophisticated and contemporary machine tools. Rotating machines are the most critical equipment in today's modern industry setups. Since induction motors are inexpensive, simple to install, and require little maintenance, they are frequently employed in industry. These equipment's operating conditions must be maintained in order to avoid production interruptions and financial loss [1]. Rotating machines are operating for a long time of duration, because of that they have been suffered from various kinds of electrical and mechanical strains. Because of these electrical and mechanical stresses, various parts of machines like bearings, gears, windings are going to fail. Various studies state that more than 50% of the total number of failures is caused by bearing related faults [2]. So, fault diagnosis of motor bearings is significant for ensuring uninterrupted machine operation. Because of this, there is a greater need for creating fault detection and fault diagnosis systems for bearing failures in industrial applications to avoid serious malfunctions and increase the dependability of the industrial machines.

Various stator, rotor and bearing faults in induction motor have been occurred because of mechanical stresses, frequent start of the motor at rated voltage, misalignment of bearings, abnormal connection and due to failure of insulation of the stator winding, over voltage, under voltage, overload, shorted rotor field winding, bearing and

gearbox failures. Inner race, outer race and ball bearing fault are frequently occurred in induction motor due to bearing failure.

Different types of fault which are occurred in induction motor can be classified as:

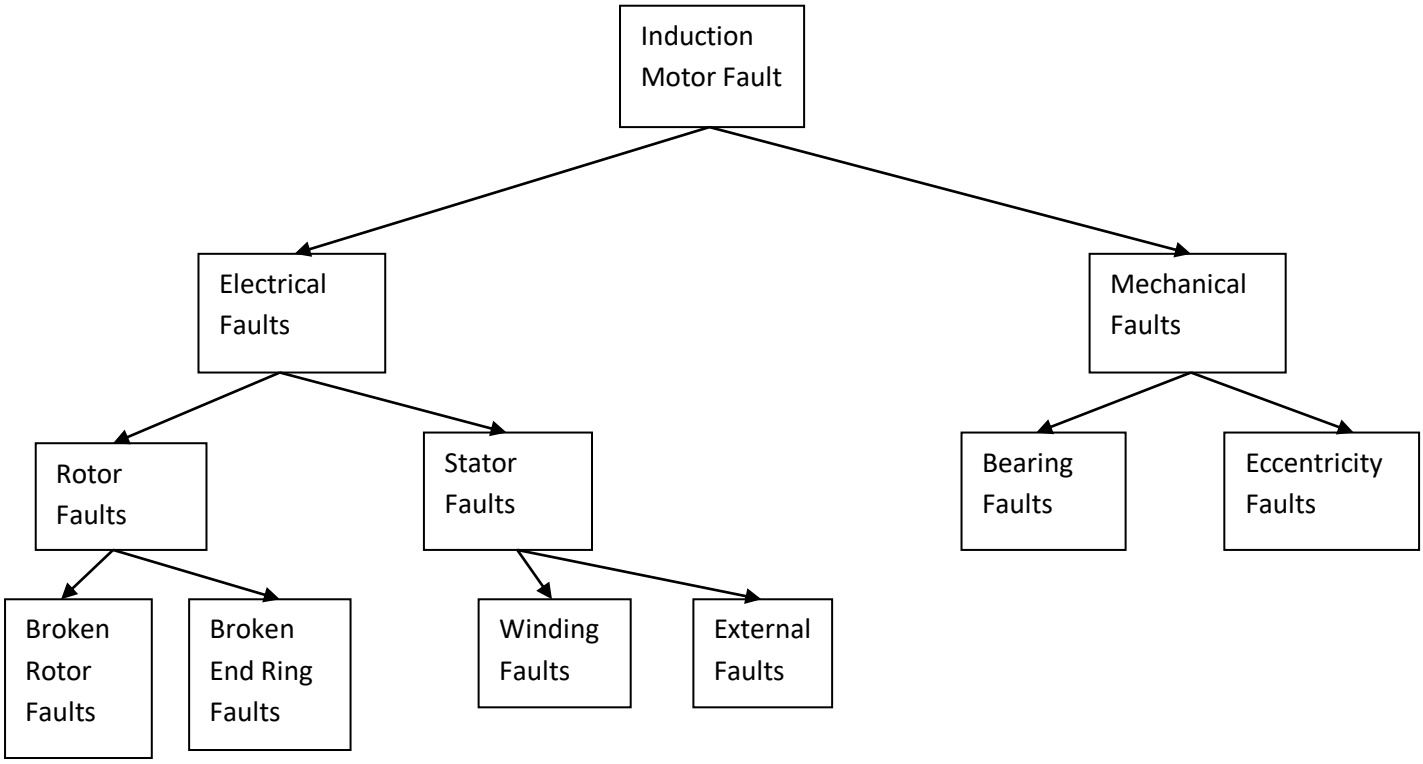


Fig 3.1. Different types of fault in Induction Motor

In recent times, Machine Learning (ML) and Deep Learning (DL) techniques are being deployed for classification of different machine faults. Because of different non linear operations in DL method, it can learn high level features to enable more thoughtful decision-making. In this chapter, outer race bearing fault has been classified using DL models. An ensemble DL model has been introduced by using weighted average method to increase the robustness of the classification. The detailed description of the dataset has been given in the section 3.3. The proposed methodology for the bearing fault classification contained three major steps, data preparation, features extraction and classification. From the vibration signal various time domain and statistical features namely maximum value, minimum value,

skewness, kurtosis, root mean square, crest factor etc have been extracted. The extracted were then fed into different deep learning algorithms for the classification. Four deep learning algorithms have been introduced in this chapter such as Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Long-short term memory (LSTM) and Gated Recurrent Unit (GRU) for the classification purpose.

The remaining of the chapter is laid out as follows: An overview of the literature review is presented in section 3.2. The description of the dataset has been mentioned in section 3.3. In section 3.4, a brief description of the proposed methodology has been mentioned. This section also contains various pre-processing steps data such as features extraction, data augmentation etc. The experimental results of different deep learning models have been introduced in section 3.5. Section 3.6 contains conclusion and future scope of the proposed work.

3.2 LITERATURE REVIEW

Rafia Nishat Toma and Alexander E. Prosvirin have proposed a methodology based on statistical features extraction and Genetic Algorithm (GA) for bearing fault classification in 2020 [2]. The researchers have extracted statistical features mean, median, variance, skewness, crest factor, Energy, kurtosis, variance etc from current signals of induction motor. Then important features were selected by using GA. Then features were fed into three different ML algorithms (KNN, Decision Tree, Random Forest) for the classification. All the three ML models have provided more than 97% of accuracy. Among these three ML models, Random Forest has provided maximum accuracy of 99.7% with a very good precision, F1-score, sensitivity and specificity.

Choug Abdelkrim and Mohamed Salah Meridje have proposed an adaptive neuro-fuzzy inference system based technique for ball bearing fault classification [3]. They have extracted five features namely mean, standard deviation, maximum, minimum and geometric mean. Then the features were fed into the adaptive neuro-fuzzy inference system for classification.

A Discrete Wavelet Transform (DWT) based method using Random Forest and XGBoost algorithm has been introduced in [4]. They have taken current signal from the motor for the classification. Then the current signal was decomposed using three wavelets. sym4, db4 and haar wavelets were used for this purpose. Then statistical features were extracted from wavelet coefficients. For training and testing, Random Forest and XGBoost were used which have provided the accuracy more than 99%. Xgboost has provided the maximum accuracy of 99.3%. Both Random Forest and XGBoost classifier have provided good AUC score of 0.99.

Pratyay Konar and Paramita Chattopadhyay have used Continuous Wavelet Transform (CWT) with SVM and ANN for bearing fault classification [5]. They have used 'morlet' and 'daubechies10' wavelet as mother wavelet. Then three features namely crest factor, kurtosis and RMS value has been extracted from the CWT coefficients for the classification. The features were then fed into SVM and ANN for classification. They have introduced different hyperparameter values c and gamma values and compared the result. They have found C= 2 and gamma= 0.2 has given the optimized result.

3.3 DATASET DESCRIPTION

The bearing dataset, which has been used in this chapter, was obtained from NASA Prognostics Data Repository. This dataset has been provided by the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati (Lee et al., 2007) [6].

This dataset contains vibration acceleration signal from the bearings. For creating this dataset, an AC motor has been used, from which a shaft was coupled. This shaft was rotating at a constant speed of 2000 RPM while having a 6000 lbs radial load mounted on it. It contained four bearings which were mounted on it and all the four bearings were force lubricated.

This dataset contains three test sets. For the first test set, two accelerometers were mounted on the bearing. For test set 2 and 3, one accelerometer has been used to get the vibration acceleration signal. All three test sets contain 2156, 984 and 4448 files respectively. Each file contains 1 second of vibration data. Sampling rate was set at 20 kHz. It states that each file has 20480 data points for all four bearings in row wise. For test set 2 and 3, the recording time interval between two 1 second vibration data was 10 minutes. For test set 1, the time interval was also 10 minutes (the first 43 files, however, were taken once every five minutes). After end of the experiment, fault has been appeared for all three test sets. For test 1, fault has been appeared in bearing 3 and bearing 4. For test set 2 and 3, an outer face bearing fault took place in bearing 2 and bearing 3 respectively. In this chapter, test set 2 has been used for outer race bearing fault classification.

3.4 METHODOLODY

In this section, data pre-processing step and classification of outer race bearing fault using DL techniques have been introduced. Data pre-processing step contains feature extraction and balance the imbalanced data. Some time domain and statistical features have been extracted from the vibration acceleration signal. Then balance the binary classification data (Normal and Outer race fault data), over sampling technique has been used on the training set. Then features were reshaped and fed into different DL algorithms for classification.

3.4.1 DATA PRE-PROCESSING

From the vibration acceleration signal which was obtained from the IMS bearing dataset, nine time domain and statistical features have been extracted namely maximum value, minimum value, mean value, standard deviation, RMS, skewness, kurtosis, form factor and crest factor. Test set 2 contains 984 files and each file consists of 20480 data points for a particular time instant. Maximum and minimum

value of 20480 points for a particular time instant have been taken as first two features. Average value of the 20480 data points for a particular time instant has been taken as the third feature (mean value). Standard Deviation is a statistical feature which can be calculated as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1}} \tag{3.1}$$

σ is defined as the standard deviation. n is the total number 20480 data points at a particular time instant. μ is the mean value of the 20480 points. Each data point is defined as x_i.

Other statistical features were skewness and kurtosis. The direction and extent of asymmetry in a normal distribution are revealed by skewness measurement. The mean, median, and mode are all the exact same in a symmetrical distribution. The degree of asymmetry or skewness increases with the distance between the mean value and the mode. There are two types of skewness present such as positive skewness and negative skewness. Skewness can be expressed as:

$$skewness = \frac{\sum_{i=1}^{n}(x_i - \mu)^3}{(n-1)\,.\,\sigma^3} \tag{3.2}$$

Kurtosis refers to the degree of flatness or peakness in the region of a normal distribution. Three types of kurtosis are present such as Lepokurtic, Mesokurtic, Platykurtic. Kurtosis can be expressed as:

$$Kurtosis = \frac{\mu_4}{\sigma^4} \tag{3.3}$$

μ_4 is defined as the fourth central moment.

RMS value is defined as:

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{3.4}$$

Form factor is defined as the ratio of RMS value and average value. It has been used as a feature for the classification. Crest factor is defined as the ratio between peak or maximum value and RMS value.

Plot of the features against time has been given below,

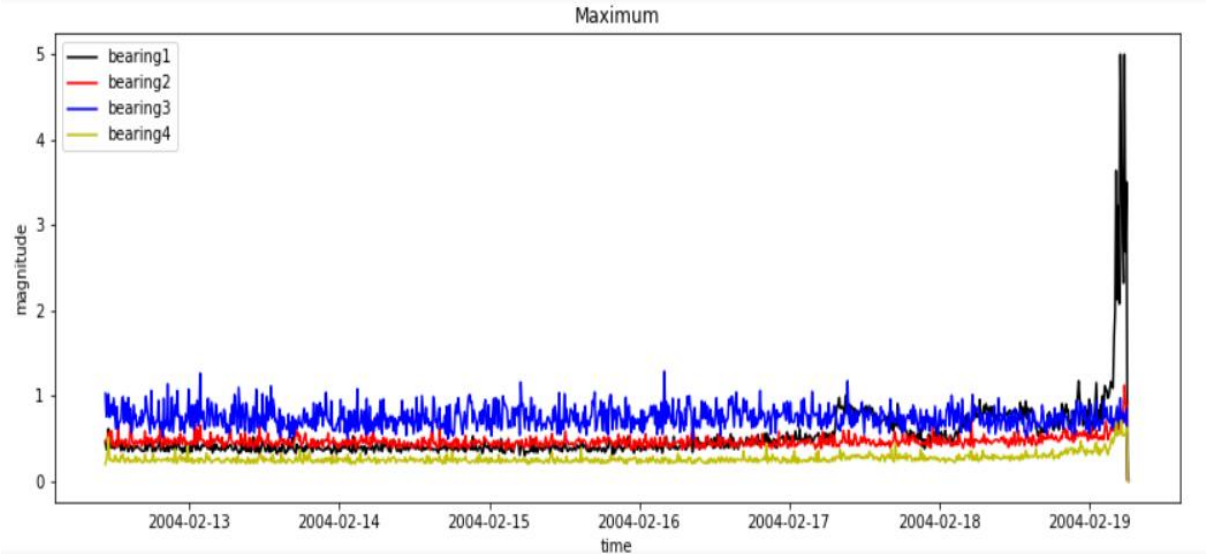


Fig 3.2. Plot between maximum value of the vibration acceleration signal for all one second time instant of four bearings and time stamp

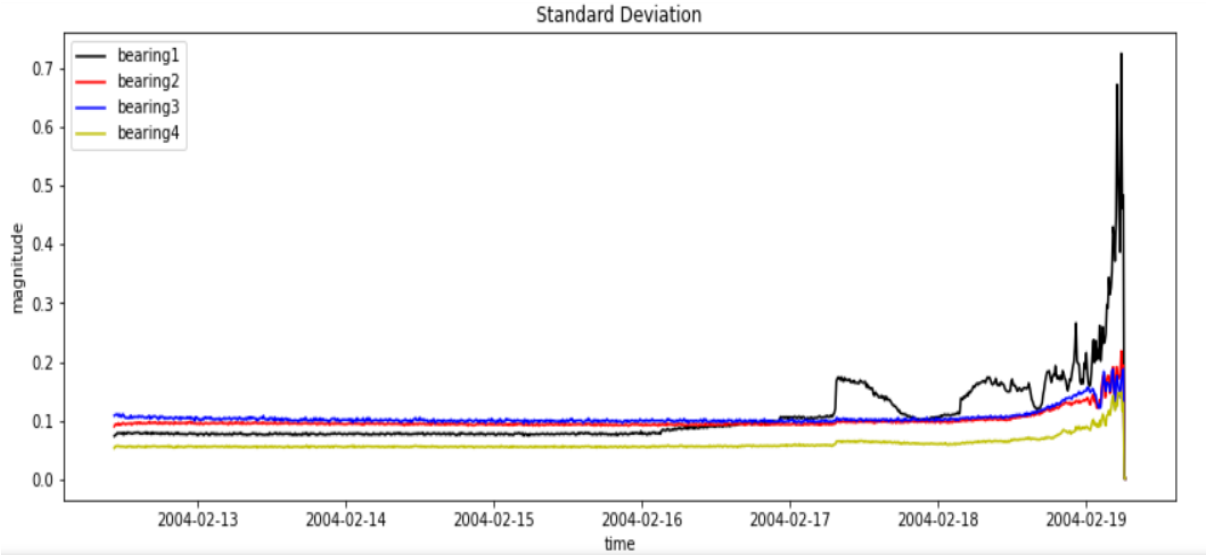


Fig 3.3. Plot between Standard Deviation of the vibration acceleration signal for all one second time instant of four bearings and time stamp

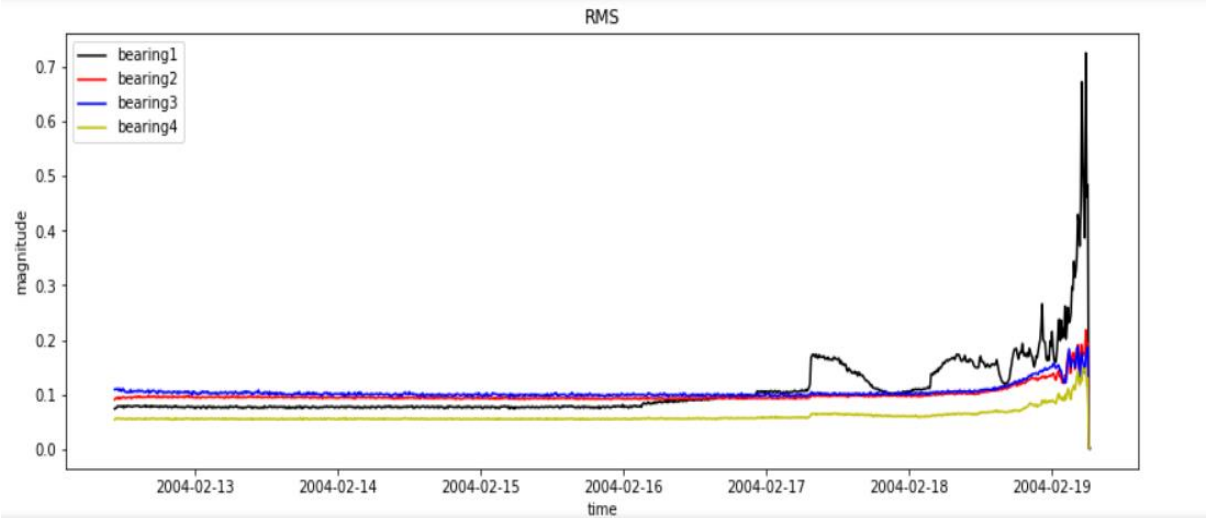


Fig 3.4. Plot between RMS value of the vibration acceleration signal for all one second time instant of four bearings and time stamp

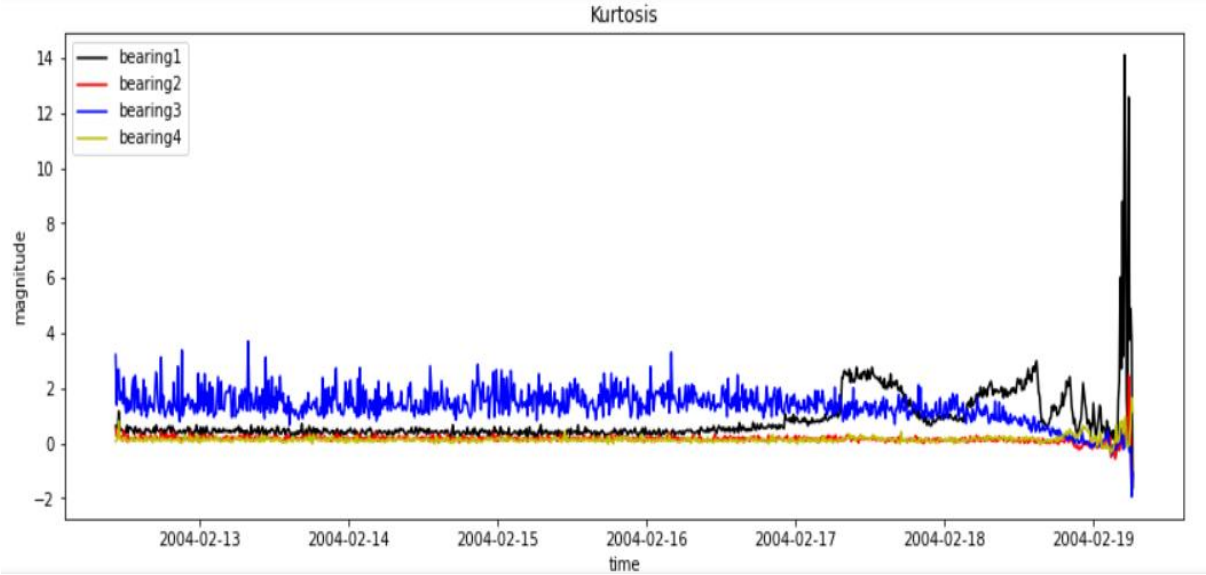


Fig 3.5. Plot between skewness of the vibration acceleration signal for all one second time instant of four bearings and time stamp

In test set 2 of the IMS bearing dataset, it has been mentioned that outer race fault has been appeared in bearing 1. From plots of the above four features, it has been shown that features were increasing abnormally in bearing 1 at the end of the experiment. So, it has been concluded that fault has been occurred in bearing 1. For each bearing, a

particular time duration (from '2004-02-13 00:02:00' to '2004-02-15 23:52:00') of vibration data have been taken as ‘normal’ label. For bearing 1, 432 files (432 data points with 9 features) have been taken for the classification. For each of bearing 2, 3 and 4, 576 files (from '2004-02-13 00:02:00' to '2004-02-15 23:52:00') have been taken as ‘normal’ labelled data. So, total 2160 files (2160 data points with nine features) were present as ‘normal’ data. This particular time duration of data has been taken as ‘normal’ label because the above feature plots did not show any abnormality in the data during this time duration. Similarly from bearing 1, total 252 files (from '2004-02-17 12:32:00' to '2004-02-19 06:22:00') of vibration acceleration data have been taken as ‘fault’ label. Fault data have been taken from the bearing 1 because outer race fault has been appeared in the bearing 1. Total 2412 data points with 9 features were present for this binary classification problem.

From 2412 data points, training set, validation set and testing set has been splitted in the ratios of 0.72, 0.08 and 0.20 respectively. For training set, 1736 data sets were present with 1536 ‘normal’ data and 200 ‘outer race fault data’. This dataset was highly imbalanced which caused the DL models biased towards the ‘normal’ class because it contained more data points. That’s why over sampling has been applied on the training set to balance the two class labels. Over sampling has been used with the over sampling ratio of 0.8. After applying over sampling, total 2764 data points were present with 1536 ‘normal’ data and 1228 ‘fault’ data. In validation dataset, 193 data points with 178 ‘normal’ and 15 ‘fault’ data were present. In testing set, 483 data points with 446 ‘normal’ data and 37 ‘fault’ data were present. Then training, validation and testing set were reshaped for the classification. Then training set was used to train the DL models. Whereas validation set and testing set were used to validate and test the models respectively.

3.4.2 CLASSIFICATION

Feature extraction technique has been discussed in previous section. Extracted features were reshaped and fed into different DL algorithms such as ANN, RNN,

LSTM, ConvLSTM, GRU. An ensembling technique has been used to imrove the model performance using weighted average method. In the subsequent sections, different DL techniques have been discussed.

3.4.2.1 ANN BASED CLASSIFICATION

In ANN, biological neurons are modelled mathematically. A biological neuron's equivalent mathematical model is shown in figure 3.6.

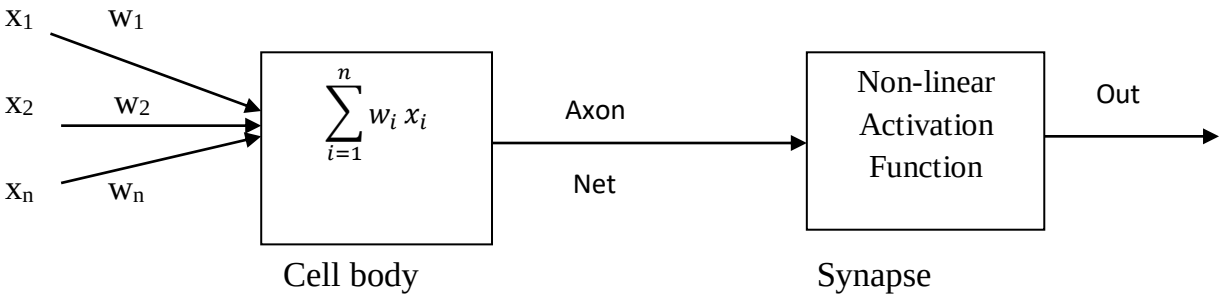


Fig 3.6. Mathematical representation of biological neuron

Weights w_1, w_2, ..., w_n denote the contribution of the inputs from the dendrons x_1, x_2, ..., x_n respectively to the overall accumulation of signal Net, where Net = $\sum_{i=1}^{n} w_i x_i$.

When the neurons are connected in a topology to transfer signal in the forward direction and there is no looping. The topology of neurons is called feedforward architecture. The neuron is represented by the following figure:

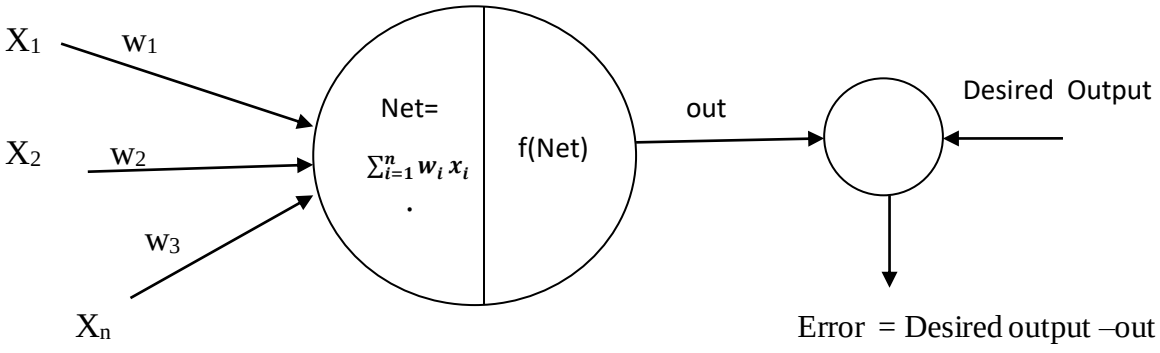


Fig 3.7. Representation of neuron in ANN

Then the feed-forward topology is represented by the following figure

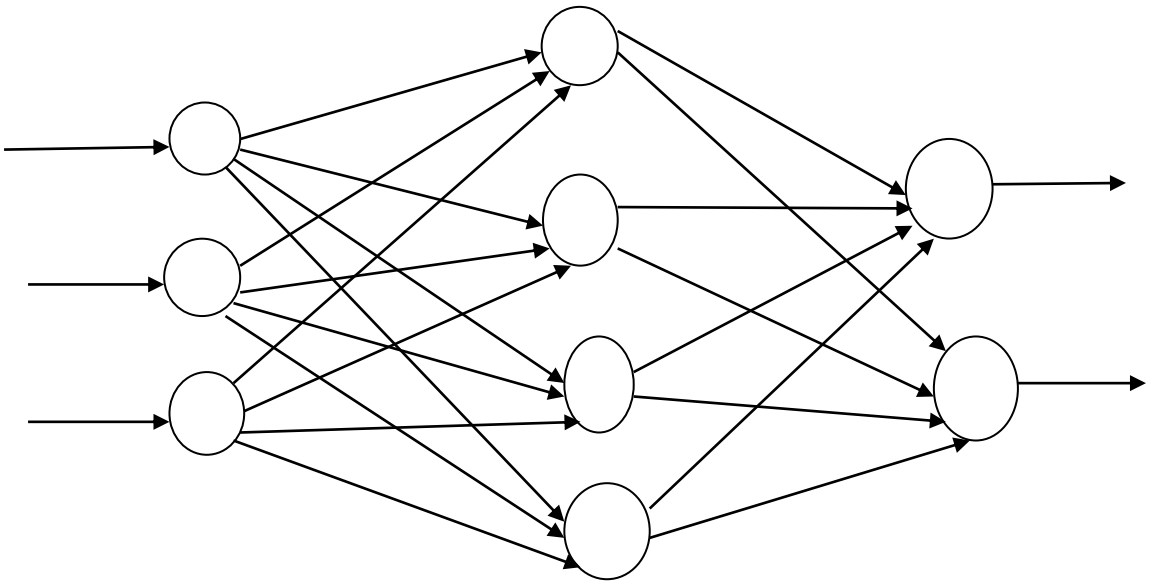


Fig 3.8. Feed-forward topology

This feed-forward topology consists of one input layer with 3 neurons, one hidden layer with 4 neurons and one output layer with 2 neurons.

For learning in ANN, an energy function is constructed,

$$E = \frac{1}{2}(d - out)^2 \tag{3.5}$$

d is the desired output and E is the energy function.

This energy function represents a parabola. So in order to minimize the error, the loss function or energy function should be present at the global minima. That is done by weight updation using back propagation algorithm. For this purpose gradient descent learning is used, which can be expressed as:

$$w_{i,new} = w_{i,old} - \eta(\partial E/\partial Wij) \tag{3.6}$$

η = learning rate which is greater than zero.

The energy surface of the weights consists of several minimas along with the global minima. While adapting the weights, if we get stucked at one of the local minima,

then obtained weights are wrong. Because also at the local minima gradient of the surface is zero. So weight adaptation will be stopped. But main objective is to reach at the bottom of the surface.

This is done by adding momentum to the weight adaptation equation, so that weight adaptation will be continuing.

The updated learning equation is

$$w_{i,new} = w_{i,old} - \eta\left(\frac{\partial E}{\partial Wij}\right) + \alpha\, \Delta wij \tag{3.7}$$

where $\Delta w_{ij} = w_{ij}(old) - w_{ij}(old\text{-}1)$ is the momentum term.

In this chapter for ANN based classification, two hidden layers have been used with 32 and 16 neurons respectively. At first the training and validation dataset with 9 features were passed through a hidden layer which contained 32 neurons followed by ReLU activation function. Now the major problem of the deep learning algorithm is to adjust itself extremely well to the training distribution, this ability is known as generalization which causes overfitting of training data. For avoiding the overfitting problem, the regularization techniques are used. That's why in this hidden layer L2 regularization have been used with a penalty factor of 0.0001. The regularization strength has been set during training by evaluating the model on the validation set. Then the activation map was passed through a dropout regularization with a dropout rate of 50%. This has been used to reduce the flexibility or variance of the model. Then the output was passed through the second hidden layer which contained 16 neurons followed by ReLU activation function and second dropout regularization with the dropout rate of 50%. L2 regularization has also been used in this layer. Then it was going to the output layer. Output layer had two neurons followed by softmax activation function for the classification. Softmax activation function has been used because it has allowed for the prediction of one class from another incompatible class in binary classification problem. For multiclass classification, softmax function converts all output scores into normalized probability distribution. For this binary class classification, softmax function can be expressed as:

$$s(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{2} e^{z_j}} \tag{3.8}$$

$s(z_j)$ is softmax function. Because it is a binary class classification problem, j is varied from 1 to 2.

One of the most used loss function in deep learning is the cross entropy loss which is a term that comes from information theory, and it is used to measure the difference between two probability distributions for a given sequence of events or random variable by comparing to the negative log-likelihood loss as a loss function. It has separate losses for binary class and multi-class. The binary cross entropy loss (LBCE) is defined as follows for binary classification:

$$LBCE(y, \hat{y}) = -\big(ylog(\hat{y})\big) + (1 - y)\log(1 - \hat{y}) \tag{3.9}$$

In this chapter, Adam optimizer has been used for updating the weights.

3.4.2.2 RNN BASED CLASSIFICATION

RNN reduces the long term dependency among the data by some extent which CNN and ANN models can't do. That's why it can be used in the prediction of sequential time-series data where the dependency among the data (present feature is dependent on past features) is present. RNN is able to capture the sequential information present in the data. In RNN, outputs of the hidden layer neurons are feedback to the input of the neurons in the hidden layer. The simplified RNN architecture has been shown in figure 3.9.

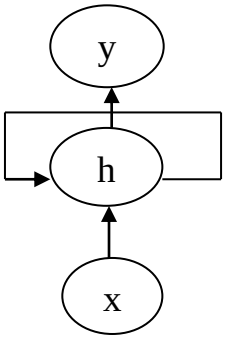


Fig 3.9. Simple architecture of RNN

The detailed architecture of RNN has been shown in figure 3.10.

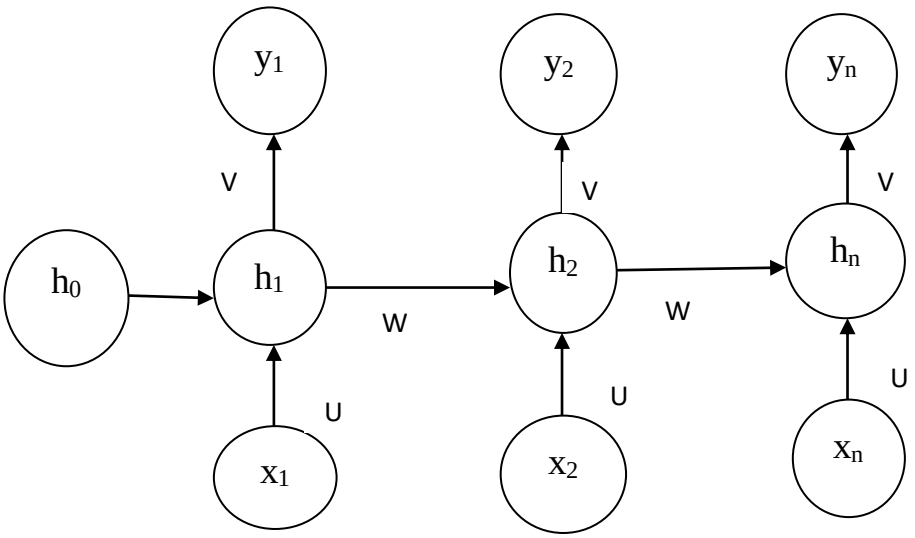


Fig 3.10. Complete architecture of RNN

From the figure it has been shown that output of hidden layer h_0 is feedback to the input of hidden layer h_1 and also output of hidden layer h_1 is feedback to the input of hidden layer h_2. Input x_1, x_2,....., x_n are the inputs at time t= 1, 2,...., n. Output sequences are y_1, y_2,....., y_n. All hidden layers (h_1, h_2,....., h_n) represent a single hidden layer with same weight and bias at different time instants of t= 1, 2,...., n. In this way, RNN is able to capture the sequential information. The output of the hidden layer and the output sequence at time t=n can be expressed as:

$$h_n = \tanh(Ux_n + wh_{n-1}) \quad (3.10)$$

$$y_n = softmax(Vh_n) \quad (3.11)$$

V is weight matrix that connects the hidden layer with the output. U is the weight matrix that connects input to the hidden layer and W are weight matrices that connect hidden layers among themselves. For updating the weight and bias of the RNN, back-propagation through time (BPTT) has been used.

For the bearing fault classification, two RNN layers, one hidden dense layer and one output layer with 2 neurons were used to train the model. At first the training set was passed through a LSTM layer which contained 150 units followed by ReLU activation

function. Then the activation map was passed through a dropout layer with a dropout rate of 50%. This dropout layer was used to reduce the over fitting problem. Then the output of this dropout layer was passed through the second LSTM layer which contained 64 units followed by ReLU activation function and second dropout layer with the dropout rate of 50%.Then it was passed through a Dense layer which contains 16 neurons followed by ReLU activation function and a dropout layer with 50% dropout rate. Each layer contained L2 Regularization (Ridge Regression) with a penalty rate of 0.0001 to overcome the over fitting problem.

3.4.2.3 LSTM BASED CLASSIFICATION

LSTM contains cell state or long-term memory which helps to overcome vanishing gradient problem. Cell state contains the previous time stamp's information. LSTM contains three gates namely forget gate, input gate and output gate. Figure 3.11 shows the architecture of LSTM [7].

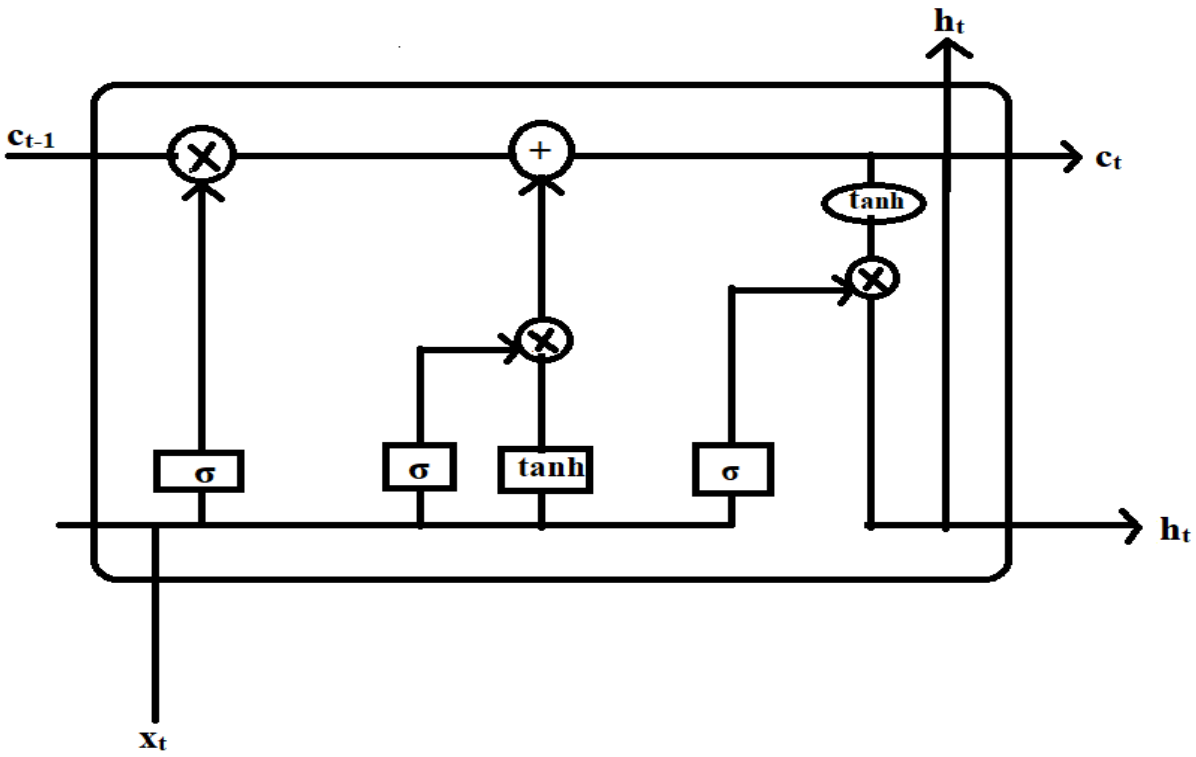


Fig 3.11. Architecture of LSTM

Forget gate decides how much information will be going to the next cell state. If forget gate output is zero, then cell state's information is cleared and if the output is 1, cell state keeps the past information. Equation of forget gate can be expressed as:

$$f_t = \sigma(W^{(f)} h_{t-1} + U^{(f)} x_t) \tag{3.12}$$

Sigmoid activation function squeezes the value between 0 and 1. f_t is forget layer output at time t and x_t is the input at time at time t. $W^{(f)}$ and $U^{(f)}$ are the weight matrix associated with h_{t-1} and x_t. h_{t-1} is the previous state's output.

Input gate decides the new data that will be kept in the cell state. It consists of two sections. Sigmoid activation function determines which values of new input data will be updated. The cell state is then updated with a vector, $\hat{c}_t$, created by a tanh layer. tanh layer converts the values between -1 and 1.

$$i_t = \sigma(W^{(i)} h_{t-1} + U^{(i)} x_t) \tag{3.13}$$

$$\hat{c}_t = tanh(W^{(c)} h_{t-1} + U^{(c)} x_t) \tag{3.14}$$

Then i_t and $\hat{c}_t$ is multiplied and then added with the c_{t-1}, which provides the updated cell state value (c_t).

$$c_t = \hat{c}_t + c_{t-1} \tag{3.15}$$

Output layer also consists of two operations. Sigmoid activation layer decides how much of information from the cell state is going to the output. Then Hyperbolic tan activation layer normalizes the cell state value between -1 and 1 and it is multiplied with sigmoid layer's output (z_t). The equations can be expressed as:

$$z_t = \sigma(W^{(z)} h_{t-1} + U^{(z)} x_t) \tag{3.16}$$

$$h_t = z_t \,.\tanh(c_t) \tag{3.17}$$

For LSTM based classification, same model architecture has been used which

was used for the RNN based classification. But in this model, instead of RNN layers two LSTM layers have been used.

3.4.2.4 GRU BASED CLASSIFICATION

GRU is a modified version of LSTM where it combines long and short-term memory into its hidden state which was proposed by (Cho, 2014) [8]. It also reduces the vanishing gradient problem which is faced by CNN, ANN and SimpleRNN network.

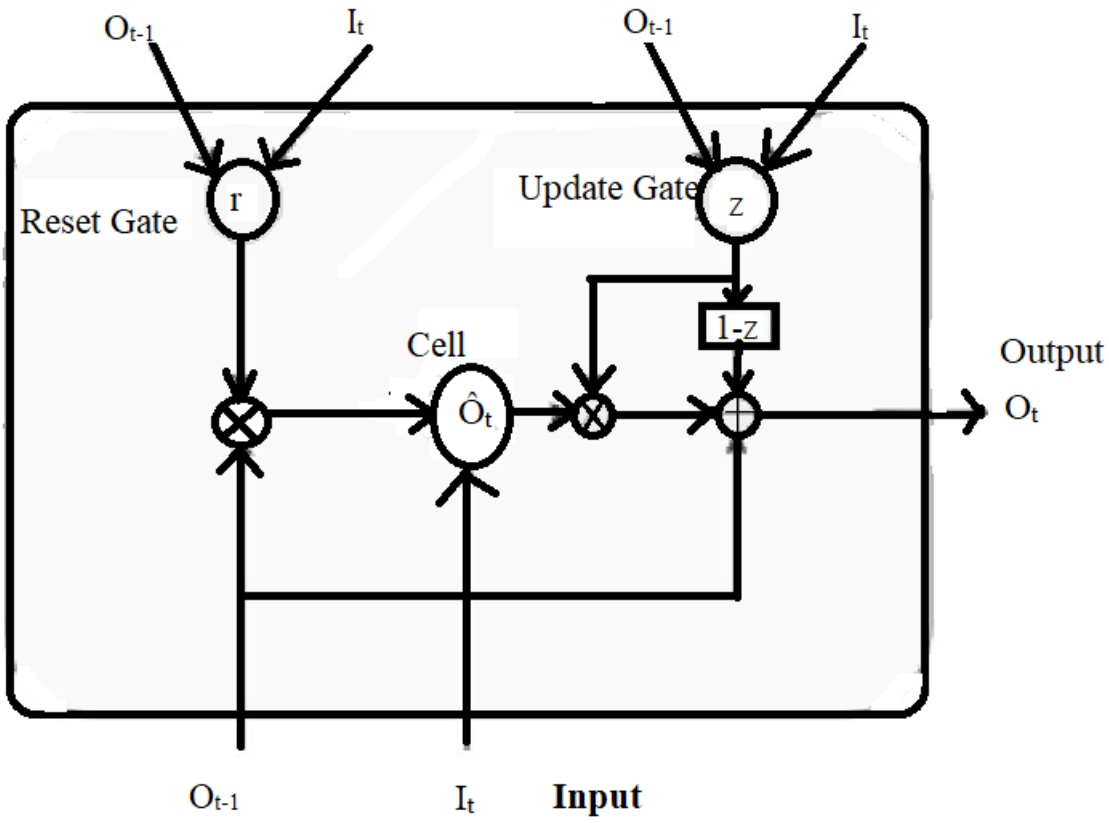


Fig 3.12. Architecture of GRU

This figure has shown the one unit of GRU network. It consists to blocks namely Update Gate and Reset Gate.

The update gate assists the GRU unit in deciding how much historical data from earlier hidden states should be transmitted to the next unit. At time t, Update Gate can be expressed by the following equation,

$$z_t = \sigma(W^{(z)} O_{t-1} + U^{(z)} I_t) \tag{3.18}$$

σ is the sigmoid activation function. $W^{(z)}$ is the weight vector associated with the hidden state's output (O_{t-1}) in Update Gate and $U^{(z)}$ is the weight vector associated with the input at time-instant t (I_t) in Update Gate.

Reset gate controls what proportion of the previous data should be forgotten in the model.. It can be expressed as:

$$r_t = \sigma(W^{(r)} O_{t-1} + U^{(r)} I_t) \tag{3.19}$$

Cell ($\hat{O}_t$) can be expressed as:

$$\hat{O}_t = tanh(U^{(r)} I_t + r_t . W^{(r)} O_{t-1}) \tag{3.20}$$

Here elementwise product is done between reset gate (r_t) and weighted hidden state's output ($W^{(r)}O_{t-1}$).

Final output is given by:

$$O_t = z_t . O_{t-1} + (1 - z_t) . \hat{O}_t \tag{3.21}$$

This final output is going to the next GRU unit.

For GRU based classification, same model architecture has been used which was used for the RNN based classification. But in this model, instead of RNN layers two GRU layers have been used.

3.4.2.5 ConvLSTM BASED CLASSIFICATION

ConvLSTM is an another variation of the LSTM model where convolution operation is done instead of matrix multiplication. In doing so, it uses convolution operations on multi-dimensional data to extract inherent spatial information. ConvLSTM takes 3-dimensional data as input.

3.4.2.6 WEIGHTED AVERAGE ENSEMBLE BASED CLASSIFICATION

In weighted average ensemble, each DL model's contribution to the classification is weighed according to the model's performance for improving the classification result. In this chapter, five DL models have been used namely ANN, RNN, LSTM, ConvLSTM, GRU. For weighted average ensemble, different weights have been chosen according to these five DL model's classification result. The total weights added up to one. As it was a binary classification problem, output layer of all the DL models has provided two prediction probabilities for two class labels. Then assigned weights were multiplied with their corresponding DL model's prediction probability. Then all five DL model's weighted prediction probabilities were summed up and argmax function was applied which provided the ensemble model's predict class label.

3.4.3 ARCHITECTURE OF THE PROPOSED METHODOLOGY

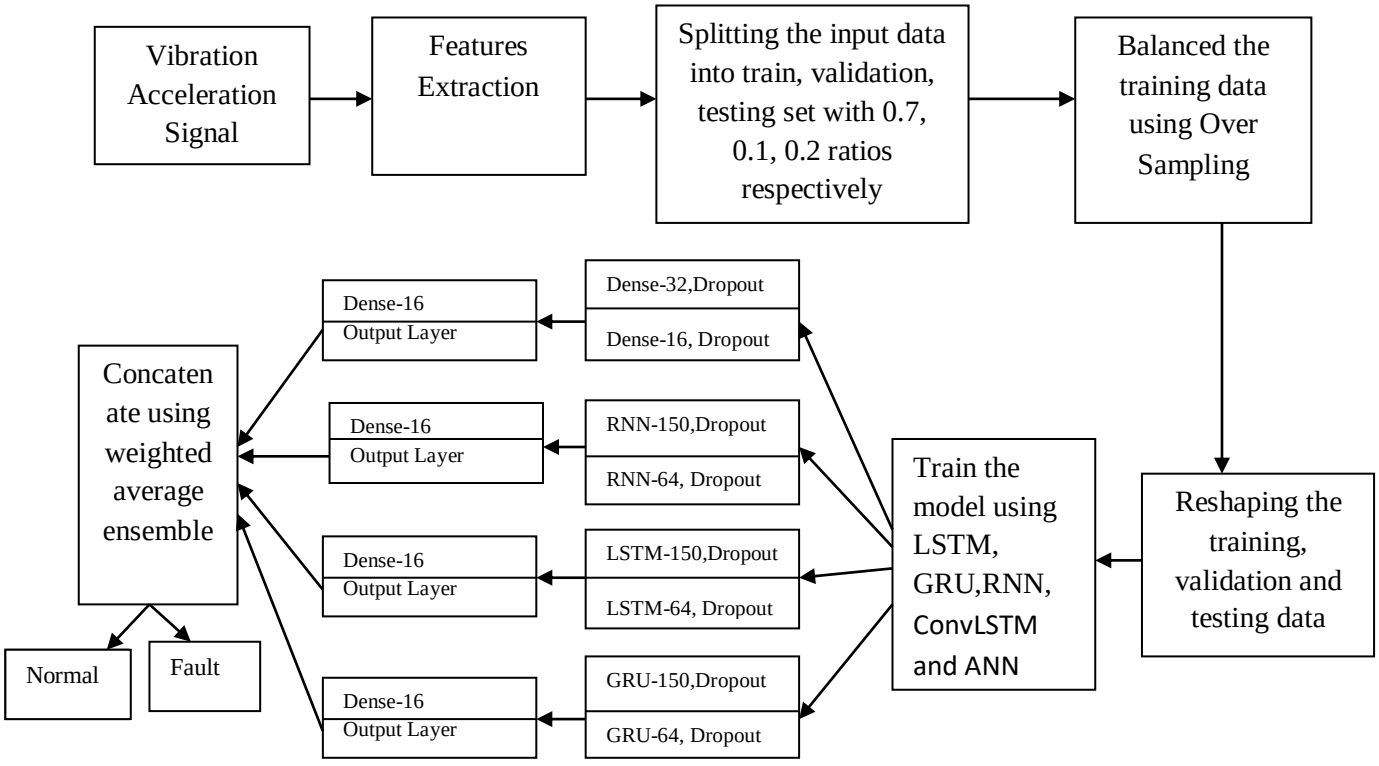


Fig 3.13. Architecture of the proposed methodology

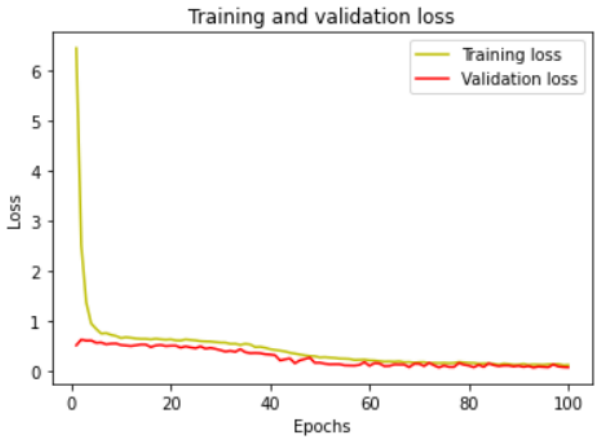
3.5 EXPERIMENTAL RESULTS

In this section, accuracy, precision, recall, f1-score and area under curve (AUC) were used to evaluate the performance of different Deep Learning classifiers. These evaluation metrics have already been discussed in section 3.5. Training and validation loss curves have been shown for all classifiers. For training the models, 100 epochs have been applied. Normal state of the machine has been taken as 'Negative' label and outer race fault has been taken as 'Positive' label.

3.5.1 ANN CLASSIFICATION BASED RESULT

ANN has provided 98.55% of accuracy with good recall, precision, f1-score of 0.92, 0.89 and 0.91 respectively. Figure 3.14(a) represents the training loss and validation loss curve vs. number of epochs. Figure 3.14(b) shows training and validation accuracy curve vs. epochs.

(a)



(b)

Fig 3.14. (a) Training and Validation loss vs. number of epochs, (b) Training and Validation accuracy vs. number of epochs

From the figure it has been shown that training and validation losses were decreasing as the number of epochs were increasing. Similarly, as the number of epochs increased, training and validation accuracy increased.

Figure 3.15 and Figure 3.16 represent the confusion matrix and Area Under the curve of this binary classification problem by using ANN.

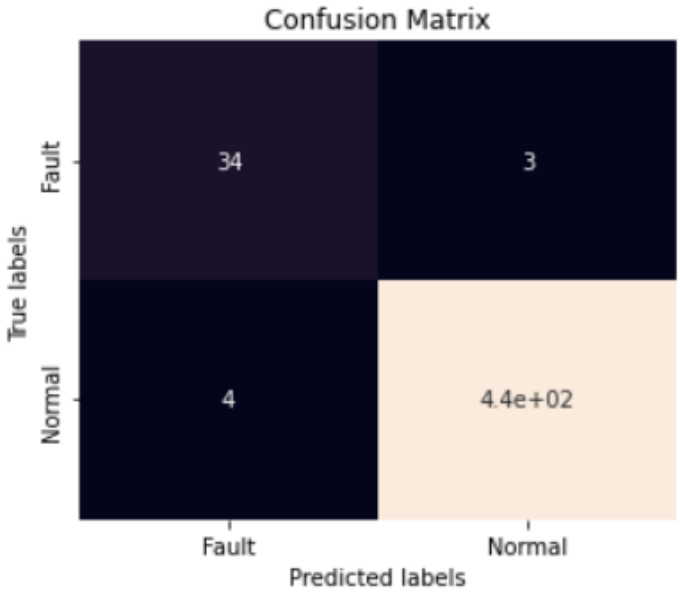


Fig 3.15. Confusion Matrix by using ANN

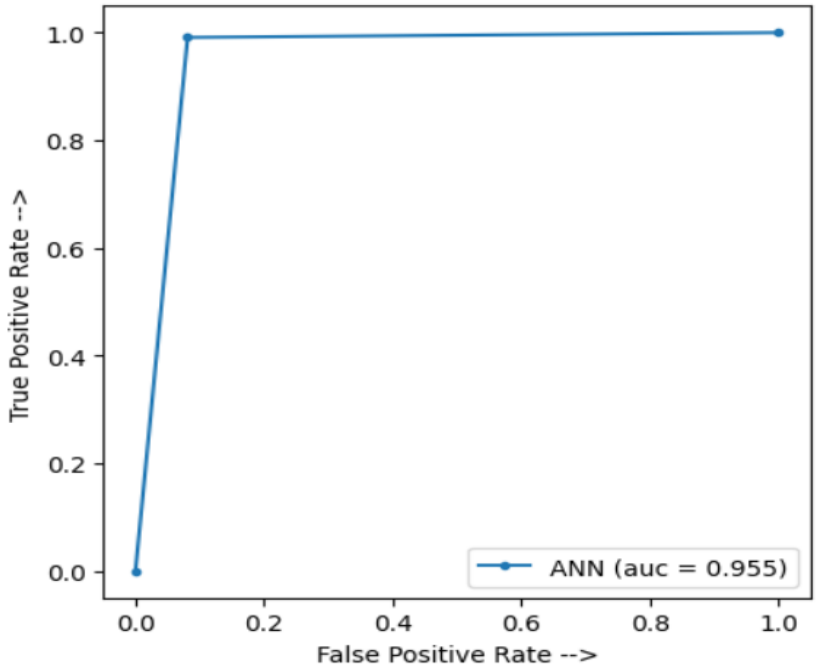


Fig 3.16. Area under Curve by using ANN

ANN has provided the AUC score of 0.955.

3.5.2 RNN CLASSIFICATION BASED RESULT

RNN has provided 96.89% of accuracy with good precision, recall, f1-score of 0.73, 0.95 and 0.82 respectively. Figure 3.17(a) represents the training loss and

validation loss curve vs. number of epochs. Figure 3.17(b) shows training and validation accuracy curve vs. epochs.

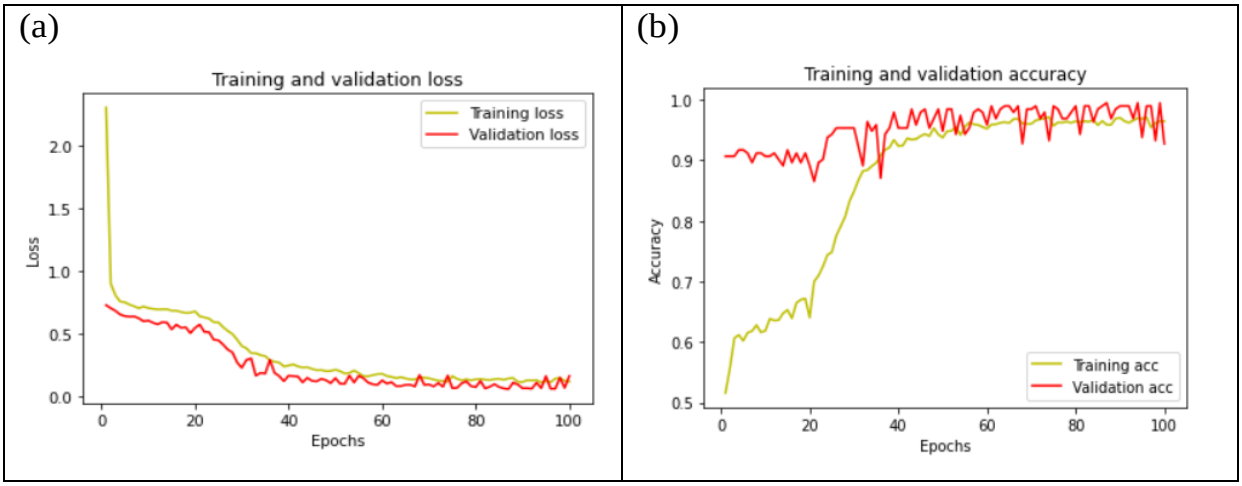


Fig 3.17. (a) Training and Validation loss vs. number of epochs, (b) Training and Validation accuracy vs. number of epochs

From the figure it has been shown that training and validation losses were decreasing as the number of epochs were increasing. Similarly, as the number of epochs increased, training and validation accuracy increased.

Figure 3.18 and Figure 3.19 represent the confusion matrix and Area Under the curve of this binary classification problem by using RNN.

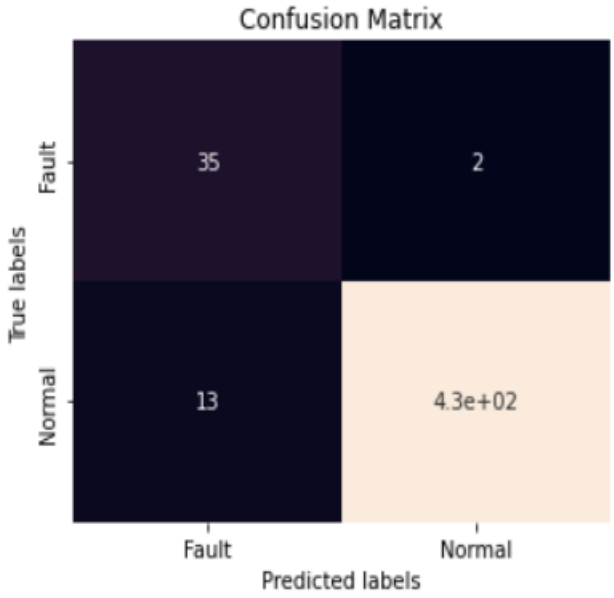


Fig 3.18. Confusion Matrix by using RNN

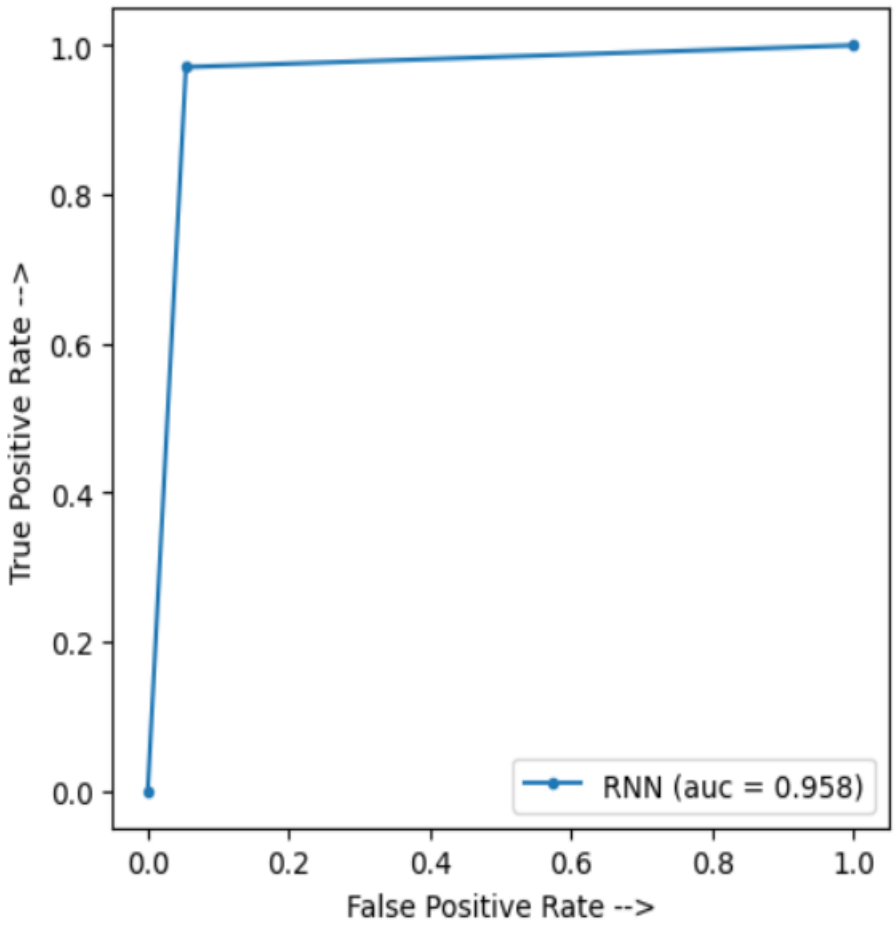


Fig 3.19. Area under Curve by using RNN

RNN has provided the AUC score of 0.958.

3.5.3 LSTM CLASSIFICATION BASED RESULT

LSTM has provided 98.55% of accuracy with good precision, recall, f1-score of 0.92, 0.89 and 0.90 respectively. Figure 3.20(a) represents the training loss and validation loss curve vs. number of epochs. Figure 3.20(b) shows training and validation accuracy curve vs. epochs.

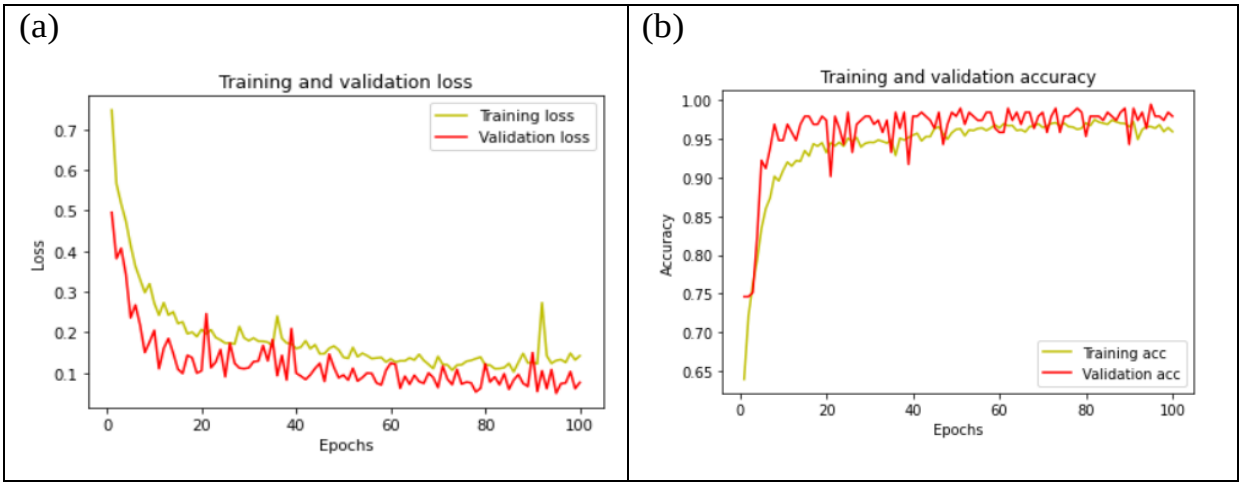


Fig 3.20. (a) Training and Validation loss vs. number of epochs, (b) Training and Validation accuracy vs. number of epochs

From the figure it has been shown that training and validation losses were decreasing as the number of epochs were increasing. Similarly, as the number of epochs increased, training and validation accuracy increased.

Figure 3.21 and Figure 3.22 represent the confusion matrix and Area Under the curve of this binary classification problem by using LSTM.

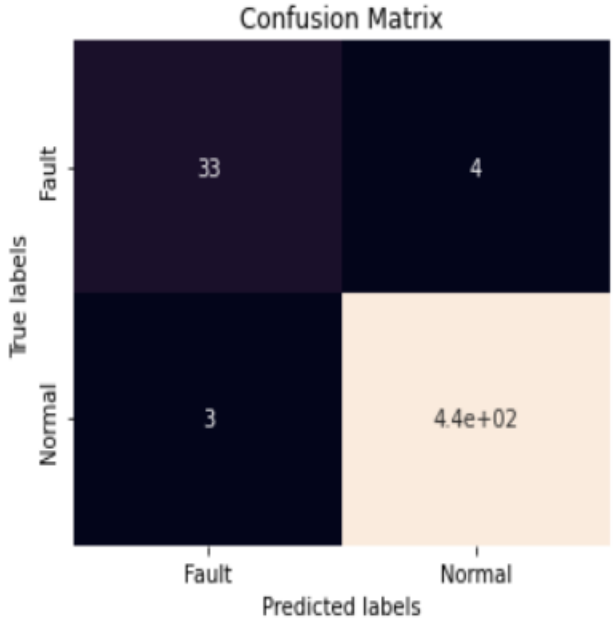


Fig 3.21. Confusion Matrix by using LSTM

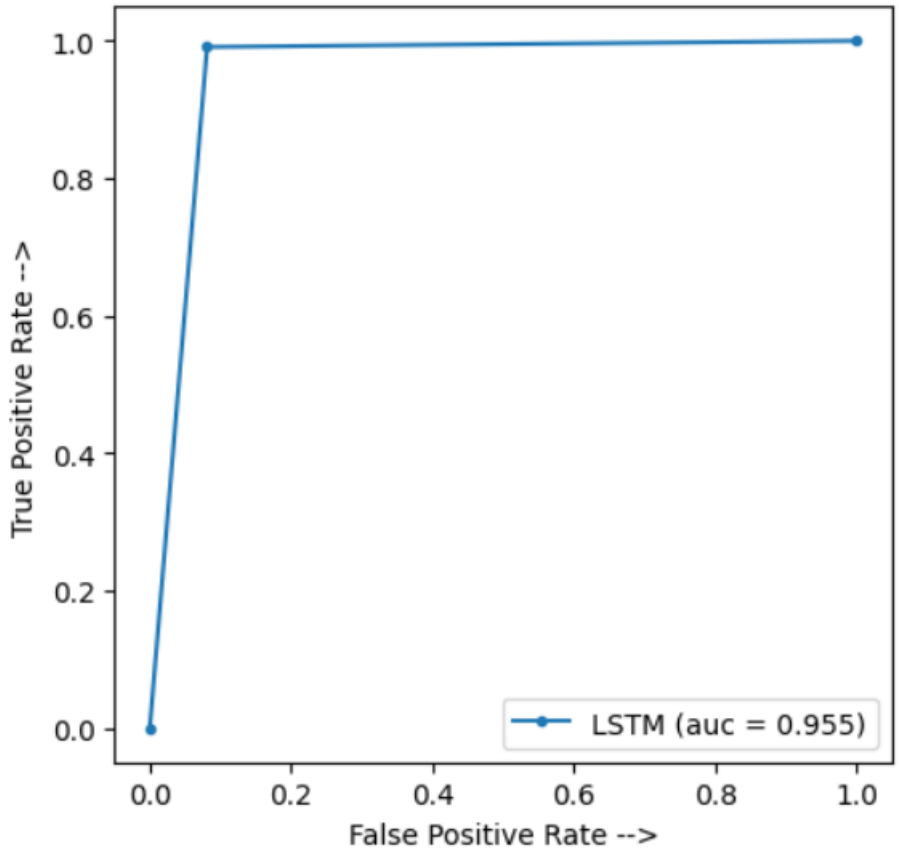


Fig 3.22. Area under Curve by using LSTM

RNN has provided the AUC score of 0.955.

3.5.4 GRU CLASSIFICATION BASED RESULT

GRU has provided 98.75% of accuracy with good precision, recall, f1-score of 0.94, 0.89 and 0.92 respectively. Figure 3.23(a) represents the training loss and validation loss curve vs. number of epochs. Figure 3.23(b) shows training and validation accuracy curve vs. epochs.

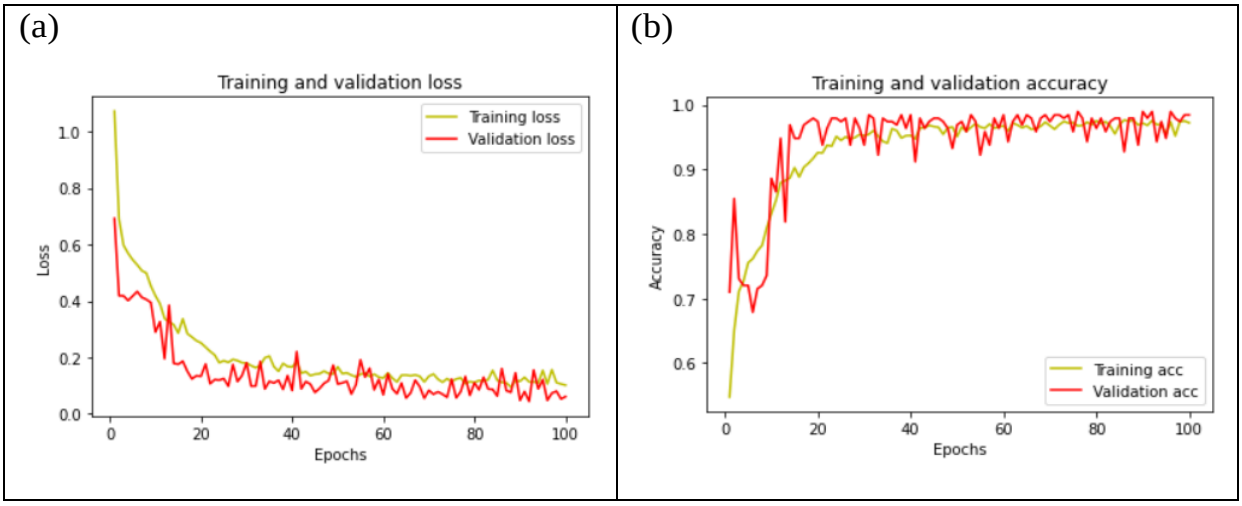


Fig 3.23. (a) Training and Validation loss vs. number of epochs, (b) Training and Validation accuracy vs. number of epochs

From the figure it has been shown that training and validation losses were decreasing as the number of epochs were increasing. Similarly, as the number of epochs increased, training and validation accuracy increased.

Figure 3.24 and Figure 3.25 represent the confusion matrix and Area Under the curve of this binary classification problem by using GRU.

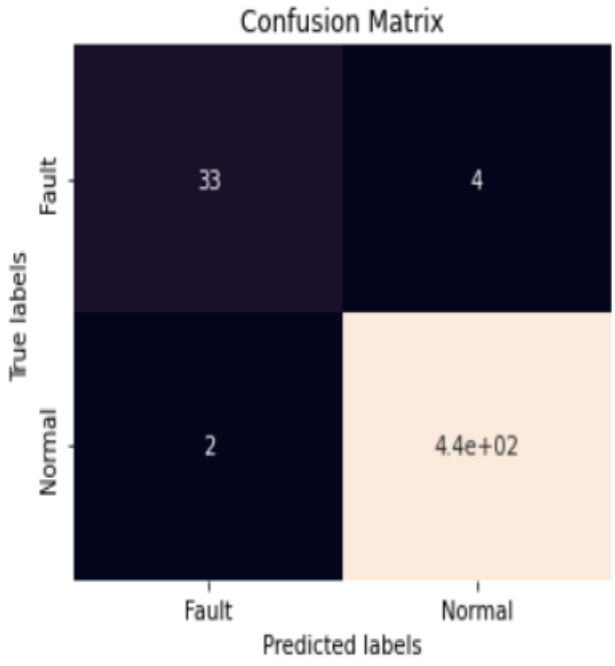


Fig 3.24. Confusion Matrix by using GRU

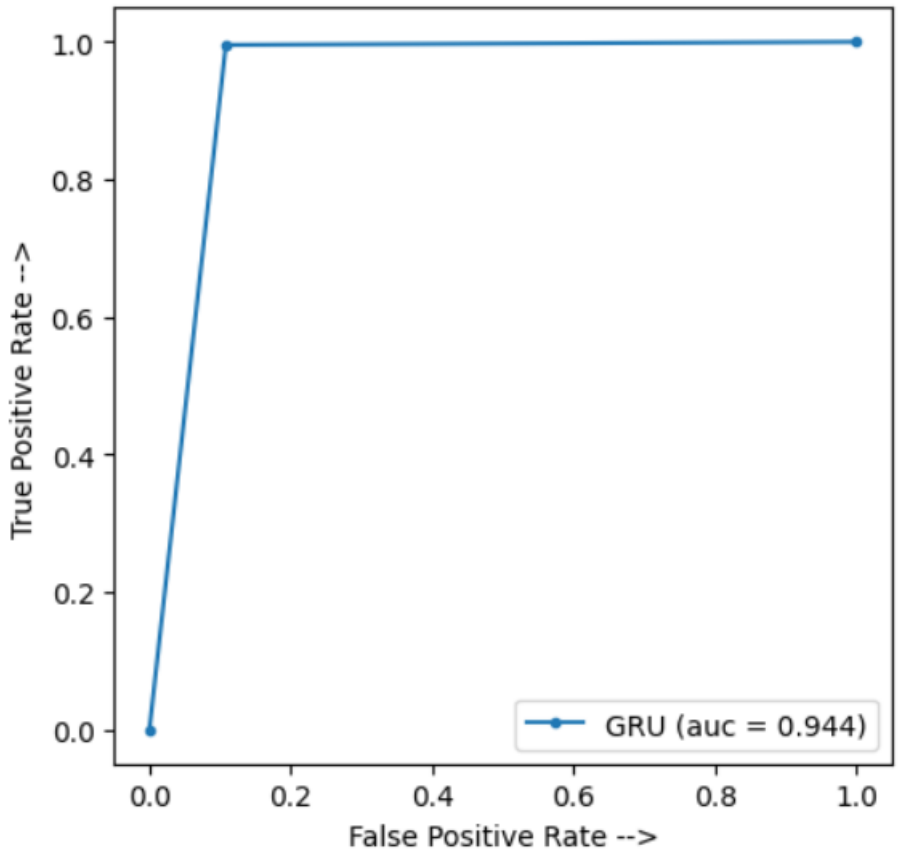


Fig 3.25. Area under Curve by using GRU

GRU has provided the AUC score of 0.944.

3.5.5 ConvLSTM CLASSIFICATION BASED RESULT

ConvLSTM has provided 98.75% of accuracy with good precision, recall, f1-score of 0.94, 0.89 and 0.92 respectively. Figure 3.26(a) represents the training loss and validation loss curve vs. number of epochs. Figure 3.26(b) shows training and validation accuracy curve vs. epochs.

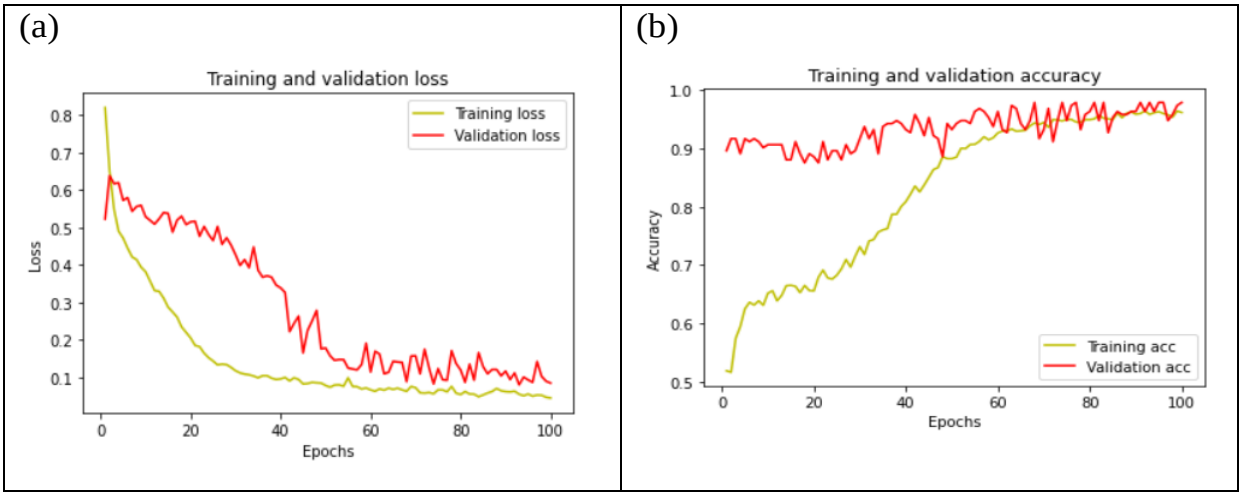


Fig 3.26. (a) Training and Validation loss vs. number of epochs, (b) Training and Validation accuracy vs. number of epochs

From the figure it has been shown that training and validation losses were decreasing as the number of epochs were increasing. Similarly, as the number of epochs increased, training and validation accuracy increased.

Figure 3.27 and Figure 3.28 represent the confusion matrix and Area Under the curve of this binary classification problem by using ConvLSTM.

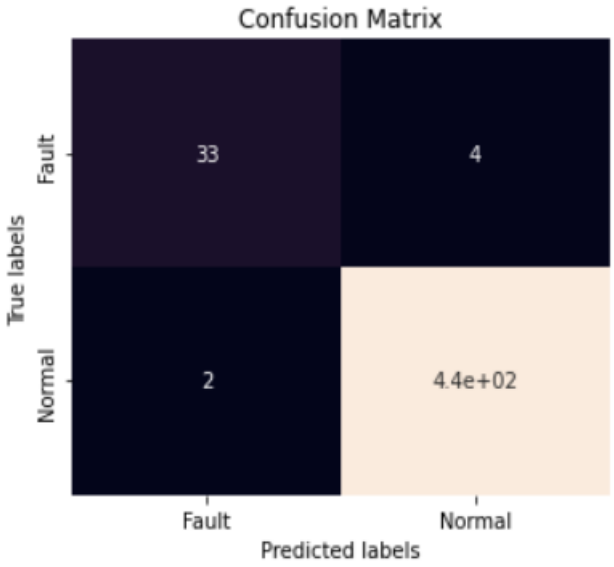


Fig 3.27. Confusion Matrix by using ConvLSTM

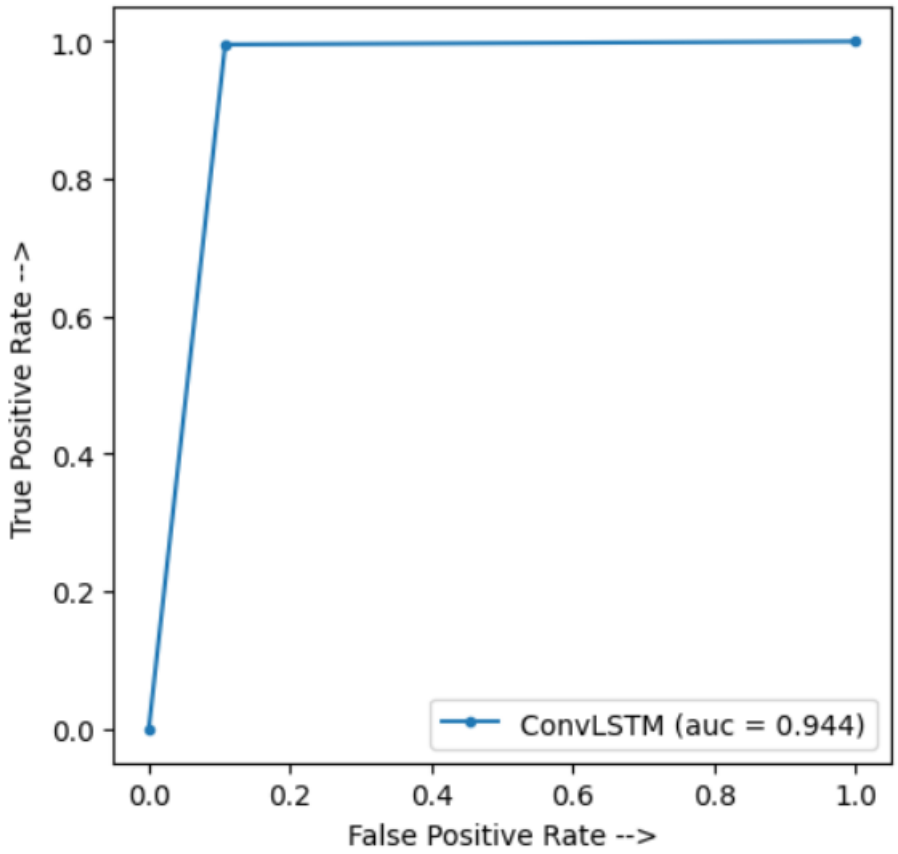


Fig 3.28. Area under Curve by using ConvLSTM

ConvLSTM has provided the AUC score of 0.944.

3.5.6 WEIGHTED AVERAGE ENSEMBLE CLASSIFICATION BASED RESULT

For different DL classifiers different weights have been assigned. Weights were assigned for ANN, RNN, LSTM, GRU, ConvLSTM were 0.10, 0.05, 0.25, 0.35 and 0.25 respectively. These weights were optimized which provided the best result using this ensemble technique. This technique has provided 99% of accuracy which was more than the accuracy of individual DL classifier. It has also provided good precision, recall and f1-score of 0.94, 0.92 and 0.93 respectively.

Figure 3.29 and Figure 3.30 represent the confusion matrix and Area Under the curve of this binary classification problem by using this ensemble technique.

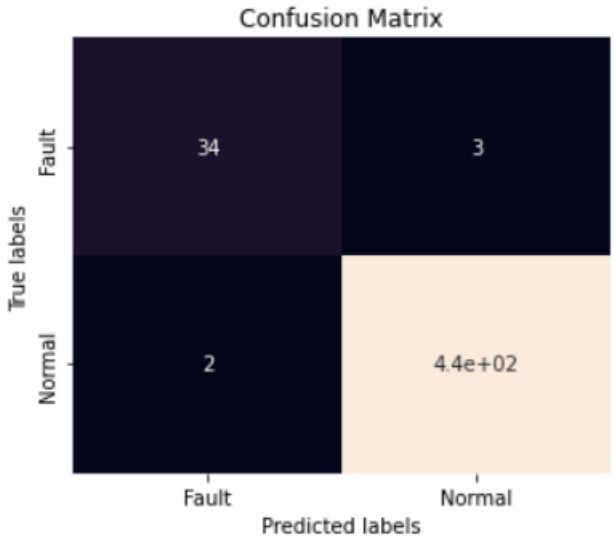


Fig 3.29. Confusion Matrix by using Weighted average ensemble

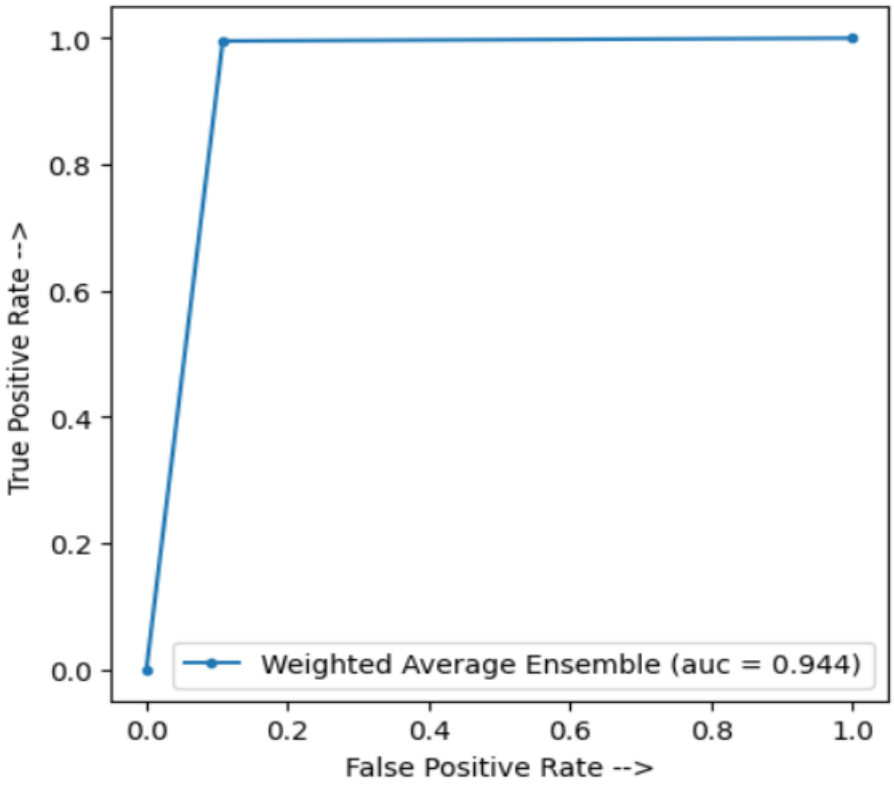


Fig 3.30. Area under Curve by using Weighted average Ensemble

This method has provided the AUC score of 0.944

3.6 CONCLUSIONS

In this chapter, vibration signals from the accelerometers have been used for bearing outer race fault classification. Some statistical and time-domain features have been extracted from the signal. Then multiple Deep Learning algorithms were implemented for classification problem. Weighted Average Ensemble technique has been further introduced in this chapter using these DL models to improve the classification performance.

Among five DL models (ANN, RNN, LSTM, GRU, ConvLSTM), GRU and ConvLSTM have provided accuracy of 98.75% with precision, recall and f1-score of 0.94, 0.89, 0.92 respectively. All the proposed DL models have provided satisfactory AUC score which was desired. For Weighted Average Ensemble technique, different weights have been assigned for the DL models according to their classification performance. These weights were optimized. This ensemble method has provided 99% of accuracy which was better than the accuracy of individual DL models. 'FN' and 'FP' were only 3 and 2 respectively. So it has been clearly stated that this ensemble technique has increased the model performance by some extent.

The proposed methodology written in this chapter can also be used for other classification problems.

References

[1] Vakharia, V., V. K. Gupta, and P. K. Kankar. "Bearing fault diagnosis using feature ranking methods and fault identification algorithms." *Procedia Engineering* 144 (2016): 343-350.

[2] Toma, Rafia Nishat, Alexander E. Prosvirin, and Jong-Myon Kim. "Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers." *Sensors* 20.7 (2020): 1884.

[3] Abdelkrim, Choug, et al. "Detection and classification of bearing faults in industrial geared motors using temporal features and adaptive neuro-fuzzy inference system." *Heliyon* 5.8 (2019): e02046.

[4] Nishat Toma, Rafia, and Jong-Myon Kim. "Bearing fault classification of induction motors using discrete wavelet transform and ensemble machine learning algorithms." *Applied Sciences* 10.15 (2020): 5251.

[5] Konar, P., and P. Chattopadhyay. "Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs)." *Applied Soft Computing* 11.6 (2011): 4203-4211.

[6] IMS bearings dataset (2014) http://ti.arc.nasa.gov/tech/dash/pcoe/prognosticdata-repository/

[7] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[8] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).

CHAPTER 4

Bearing Fault Predictive Maintenance using Deep Learning Techniques

---

4.1 INTRODUCTION

Various maintenance strategies are used in industries to achieve almost zero downtime. This primarily entails predicting and managing the condition of parts like bearings since they are responsible for the equipment failures, particularly in induction motors [1]. Because of operating for a long time of duration, the rotating machines eventually break down after a certain period of time, unless it is being maintained. Industries follow different maintenance programs to increase the operational reliability and reduce costs. There are mainly three types of health management strategies are being followed namely reactive maintenance, preventive maintenance and predictive maintenance. In reactive maintenance, the machine is used to its limit and repairs are performed only after machine failure. It costs severe damage to the machine and also costs accidents. For a complex system such as Aircraft engine, the risk can't be taken for running it to failure, as it will be extremely costly to repair. That's why many industries try to prevent failure before it occurs by performing regular checks on the machine. One big challenge with preventive maintenance is determining when to do maintenance. Since it does not know that when the failure occurs and by scheduling maintenance very early, machine life is wasting. That's why predictive maintenance is usually used. In predictive maintenance, the failure time can be predicted and maintenance can be scheduled right before the failure which will reduce the maintenance cost. This way, the downtime can be minimized and equipment lifetime is maximized [2].

In this chapter for predictive maintenance of bearing, the same dataset has been taken which was used for the outer race bearing fault classification in chapter 3. The detailed description of the dataset has been mentioned in section 3.3. This predictive maintenance problem was based on multivariate time series prediction. For this purpose, four features have been extracted from the vibration acceleration signal of the dataset. These fours features are maximum, standard deviation, kurtosis and RMS value. RMS value has been taken as the condition indicator of the machine because it was analogous with the machine's failure. With increasing RMS value, machine's degradation has been increased. All the features were converted into time-series data. In this chapter past seven cycles time-series features have been used to predict the next value of RMS or condition indicator. Before the prediction, all the features have been smoothed by non-parametric kernel density based estimation. Then LSTM [3] has been used for this time-series prediction problem.

4.2 DATASET DESCRIPTION

For this predictive maintenance problem, the same dataset has been used which was used for the classification problem in the previous chapter. The detailed description of the dataset has been introduced in section 3.2.

4.3 METHODOLOGY

For this predictive maintenance problem, four features namely Maximum, Standard Deviation, RMS and kurtosis value have been extracted from bearing 1 of test set 2 in the dataset. These features were analogous with the bearing's degradation as time increased. RMS value has been taken as health indicator. From the figure 3.1, 3.2, 3.3 and 3.3, it has been clearly shown that these four features were increasing abnormally at the end of run to failure experiment of the dataset. It has also been mentioned in the dataset description that after the experiment outer race fault has been occurred in

bearing 1 of test set 2. Bearing 1's vibration acceleration data has been taken to train and validate LSTM. LSTM has been discussed in section 3.4.2.3. For testing, bearing 2 of test set 3 has been used because it was completely at normal condition during the experiment. The main objective of this chapter was to test the model on bearing 2 of test set 3 and show that RMS health indicator was also given the good prediction about the bearing's failure in the future. For smoothing the training, validation and testing data, non-parametric kernel density based estimation has been used. It has also been used in the comparison study against the LSTM based prediction.

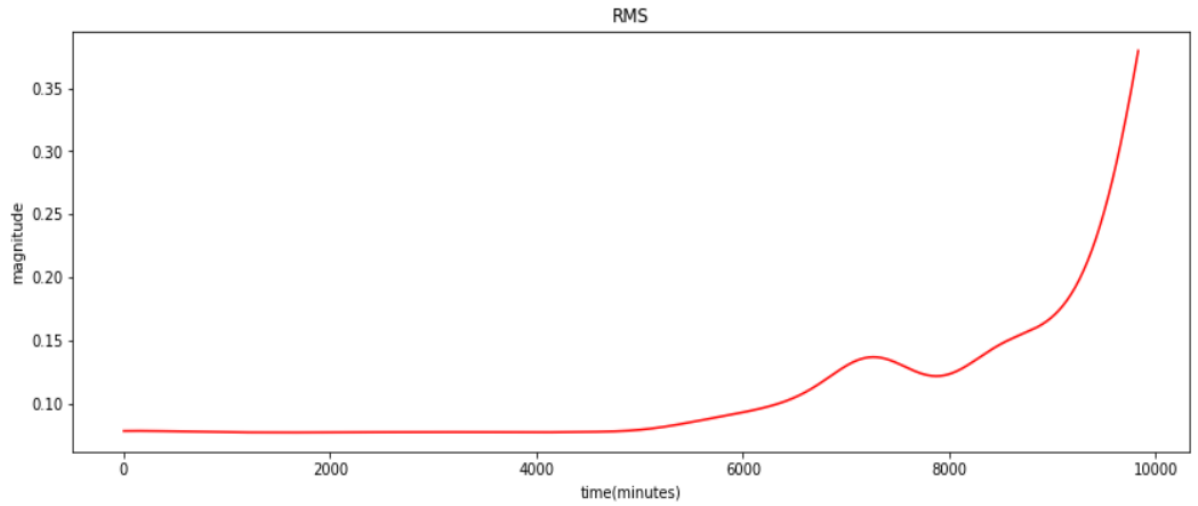


Fig 4.1. RMS health indicator after smoothing

In the figure date-time format has converted into 'minutes'.

Feature normalization has been used to normalize the features of training, validation and testing set between 0 and 1. Feature normalization was used to make all the features in the same scale, so all the features had same importance. Because of it, the optimization algorithm converges faster. Another reason is that most of the deep learning algorithms use Euclidean distance to measure the distance between two data points. It is expressed as:

$$x_{n=}\frac{x-x_{min}}{x_{max}-x_{min}}$$

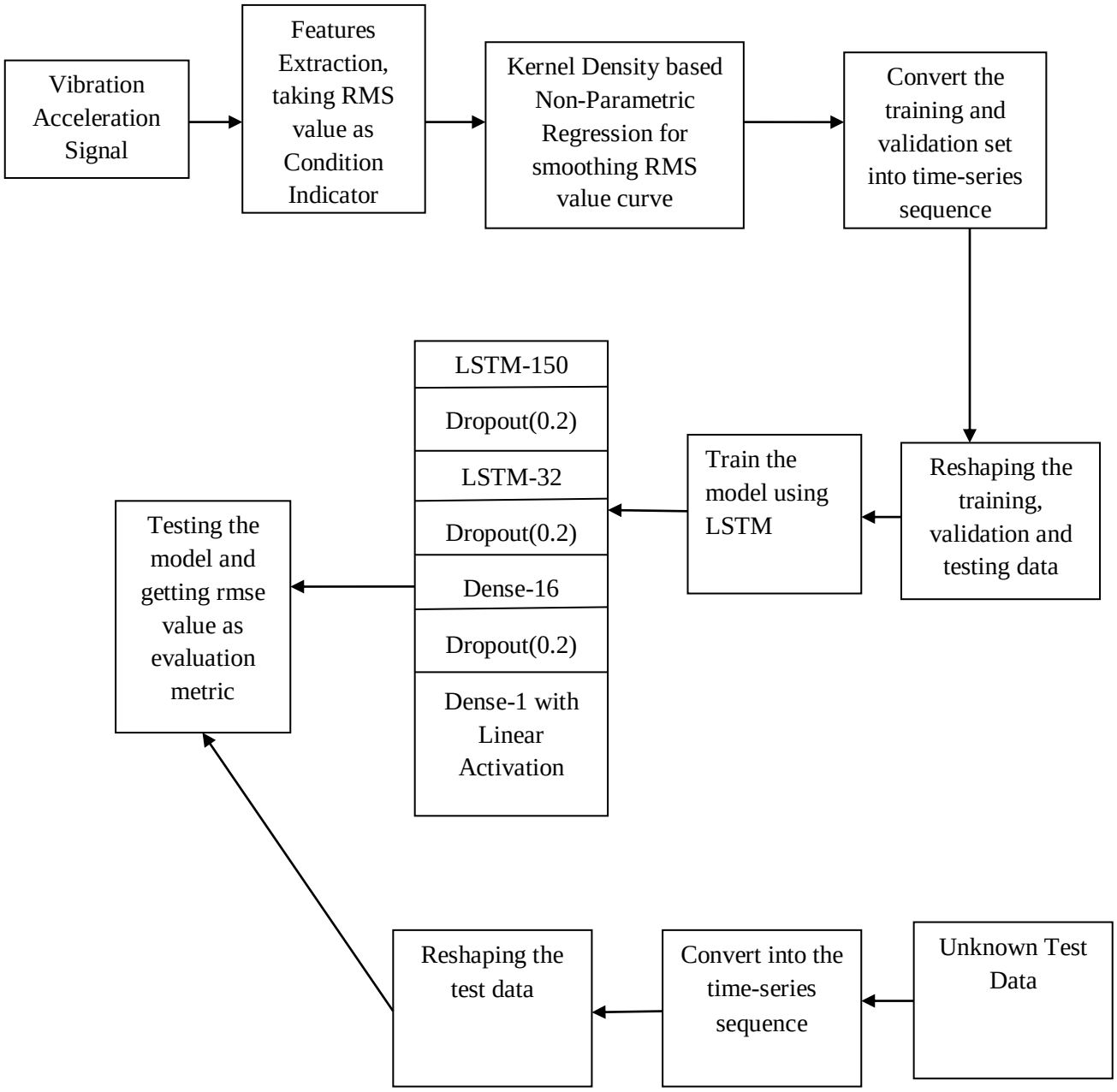
x_n is the normalized feature.

As it was a multivariate time-series forecasting problem, first 90% data of bearing 1 of test set 2 were taken as training set and next 10% data were taken as validation set. For testing, same four features from bearing 2 of test set 3 have been used for this prediction problem.

Then all training, validation and testing set have been converted into time-series data (time dependent). In this chapter past 7 data with 4 time dependent features have been used to predict the next 8^{th} time stamp's RMS value. As RMS value has been used as the condition indicator, so its value has been predicted in the next time stamp. Then next 2^{nd} to 8^{th} time stamp's data have been used to predict the 9^{th} time stamp's RMS value. It was a multivariate time forecasting problem because multiple features have been used to predict the next time stamp's condition indicator value.

In this chapter, two LSTM layers, one hidden dense layer and one output layer with 1 neuron were used to train the model. At first the training set were passed through a LSTM layer which contained 100 units followed by ReLU activation function. Then the activation map was passed through a dropout layer with a dropout rate of 20%. This dropout layer was used to reduce the over fitting problem. Then the output of this dropout layer was passed through the second LSTM layer which contained 32 units followed by ReLU activation function and second dropout layer with the dropout rate of 20%.Then it was passed through a Dense layer which contains 16 neurons followed by ReLU activation function and a dropout layer with 20 percent dropout rate. Each layer contained L2 Regularization (Ridge Regression) with a penalty rate of 0.0001 to overcome the over fitting problem. As it was a regression problem linear activation function was used at the output layer. Mean squared error has been taken as the loss function and Adam optimizer has been used for optimization. Output layer had one neuron followed by 'linear' activation function for this prediction problem.

4.3.1 ARCHITECTURE OF THE PROPOSED METHODOLOGY



4.4 EXPERIMENTAL RESULTS

Predictive maintenance of bearing was a regression problem, that's why root mean square error (rmse) has taken as evaluation metric. rmse should be low.

Consider d_1, d_2, d_3,……, d_n are the n desired outputs of a prediction problem and y_1, y_2, y_3,……, y_n are the predicted outputs of a model corresponding to the inputs x_1, x_2, x_3,……, x_n. Then rmse value can be defined as:

$$rmse = \sqrt{\frac{(y_1 - d_1)^2 + (y_2 - d_2)^2 + \cdots + (y_n - d_n)^2}{n}} \tag{4.1}$$

Training and validation loss curve has also been shown in the result. Kernel ridge regression has been introduced in the result to compare with the proposed LSTM based model. For training the LSTM model 50 epochs have been considered.

4.4.1 LSTM BASED RESULT

Training and validation loss curve has been shown by the figure 4.1.

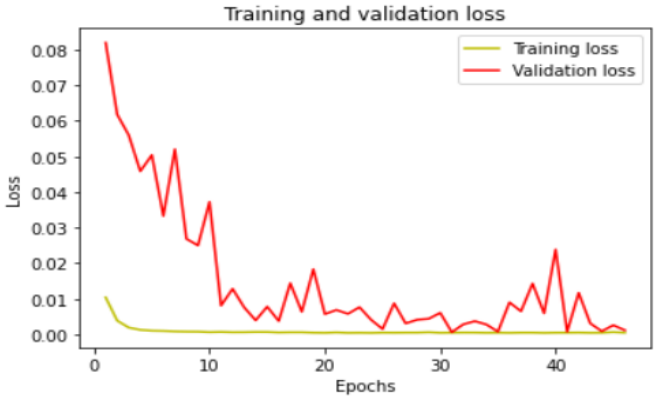


Fig 4.2. Training and Validation loss curve vs. number of epochs using LSTM

Training and validation loss were decreasing with increasing number of epochs which was desired.

Figure 4.3 represents true and predicted health indicator (RMS) curve based on the training dataset.

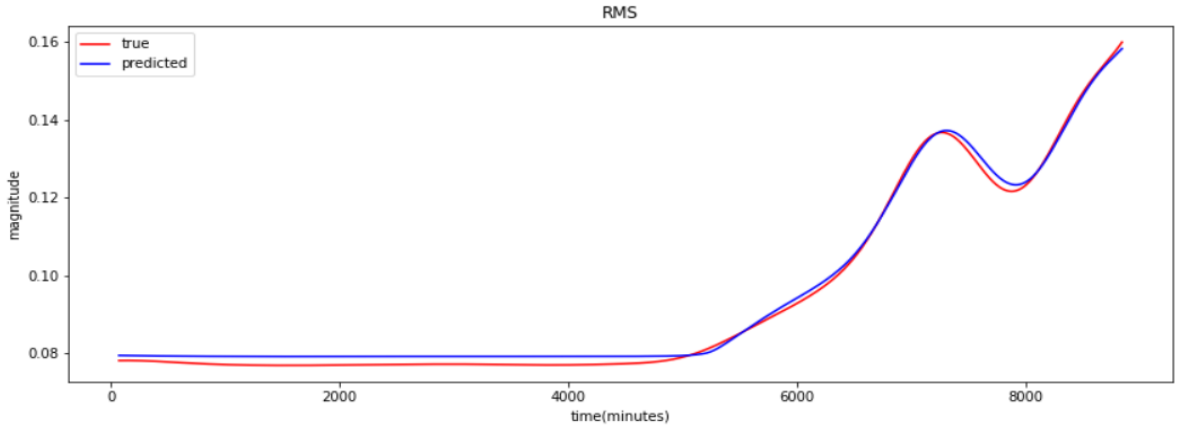


Fig 4.3. True and Predicted RMS curve based on training dataset

Train rmse score was 0.0017. From the figure, it has been shown that predicted health indicator (RMS) on the training dataset has almost followed the true health indicator (RMS) curve.

Figure 4.4 demonstrates true and predicted health indicator (RMS) curve based on the validation dataset.

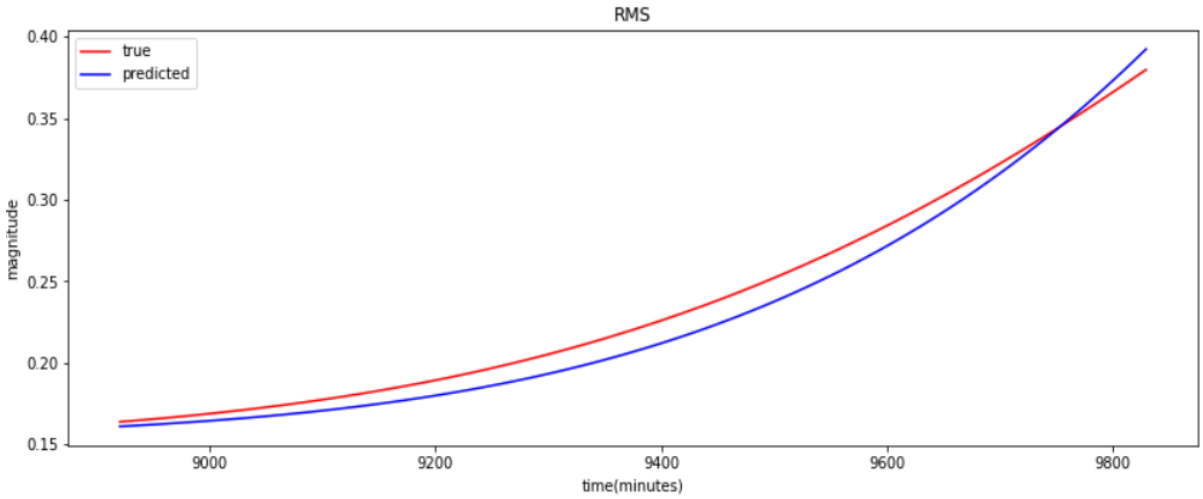


Fig 4.4. True and Predicted RMS curve based on validation dataset

Validation rmse score was 0.0101.

Figure 4.5 demonstrates true and predicted health indicator (RMS) curve based on the test dataset. Test dataset was taken from test set3 of bearing 2.

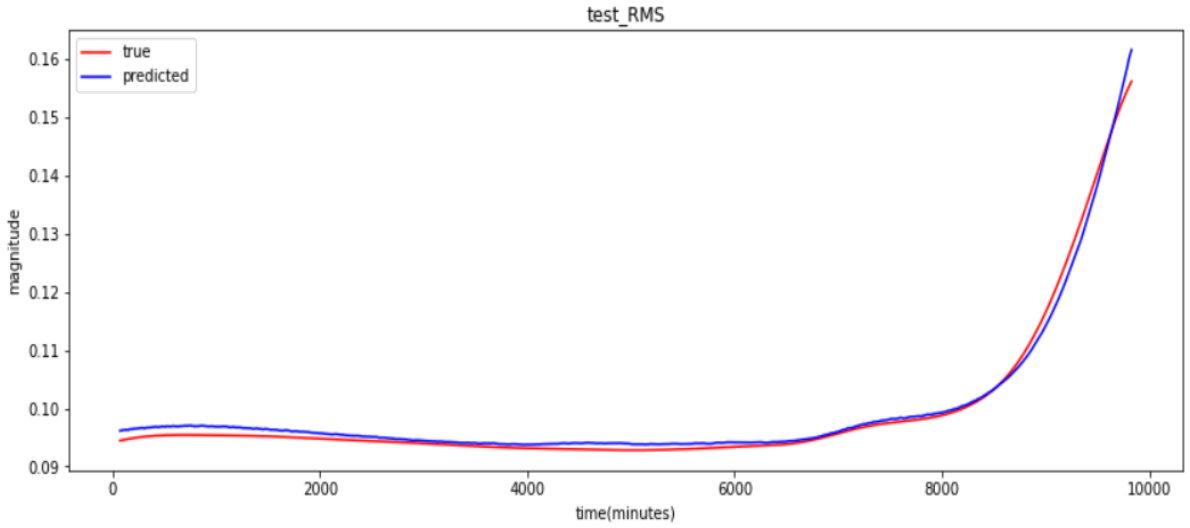


Fig 4.5. True and Predicted RMS curve based on test dataset

Test rmse score was 0.0012. It has been stated that the LSTM model performed well on the unknown test dataset also.

4.4.2 KernelRidge Regression BASED RESULT

KernelRidge Regression [4] is a supervised machine learning algorithm. It is used for regression problem. It uses both kernel trick and l2 regularization.

In this chapter, it has been used to compare with proposed LSTM model. Regularization strength has been taken as 0.1. Figure 4.6 represents true and predicted health indicator (RMS) curve based on the testing dataset dataset using KernelRidge regression. To train this model, three features (Maximum value, Standard Deviation and Kurtosis) have been taken as independent features and RMS value (health indicator) has been taken as the dependent variable.

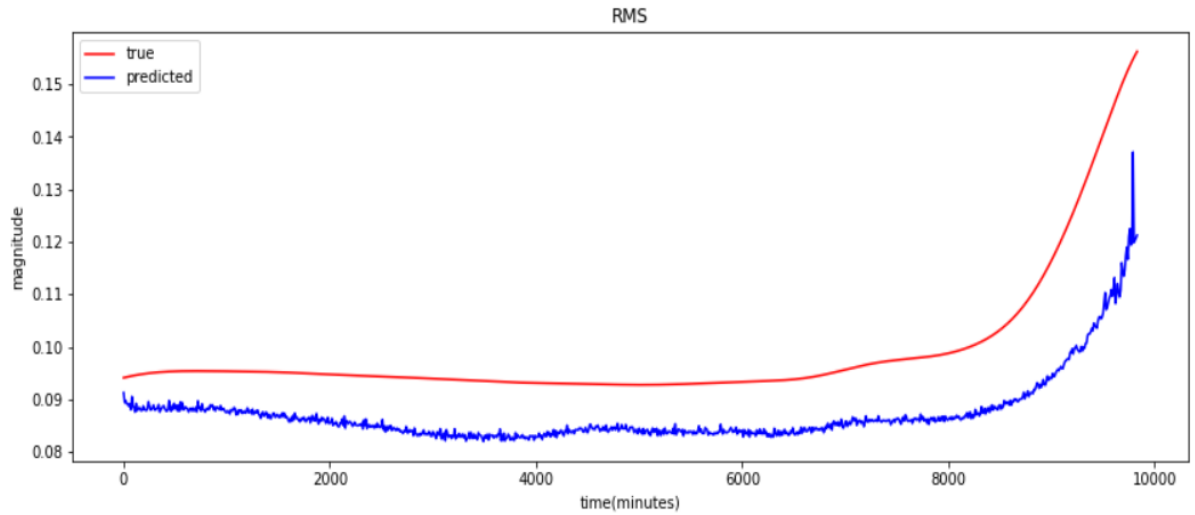


Fig 4.6. True and Predicted RMS curve based on test dataset using KernelRidge Regression

rmse score for this technique was 0.0134.

4.5 CONCLUSIONS

In this chapter, vibration signal which was used for the classification problem in chapter 3 has been introduced for the predictive maintenance of the bearing. Four statistical and time-domain features have been extracted from the signal, which were used for the predictive maintenance. Then RMS value was taken from the extracted features as condition indicator of the bearing. RMS value was analogous with machine's degradation because it has already been shown that RMS value was increasing rapidly as the fault has been occurred in the bearing. As it was a multi-variate regression problem, the features were converted into time-series data. Then LSTM has been deployed to provide this regression result. KernelRidge regression has also been introduced in this chapter to compare with LSTM.

From the experimental result, it has been clearly stated that LSTM based model has provided very less training, validation and test rmse score of 0.0017, 0.0101 and 0.0012 respectively. In the other hand, KernelRidge regression has provided

test rmse score of 0.0134 which was much bigger than LSTM based test rmse score (0.0012). Also predicted RMS condition indictor did not follow the true RMS condition indicator curve. So it has been said that the proposed LSTM based model was better than the KernelRidge regression based prediction.

References

[1] Thorsen, Olav Vaag, and Magnus Dalva. "Failure identification and analysis for high-voltage induction motors in the petrochemical industry." *IEEE Transactions on Industry Applications* 35.4 (1999): 810-818.

[2] Saxena, Abhinav, et al. "On applying the prognostic performance metrics." *Annual Conference of the PHM Society*. Vol. 1. No. 1. 2009

[3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[4] Hoerl, Arthur E., and Robert W. Kennard. "Ridge regression: Biased estimation for nonorthogonal problems." *Technometrics* 12.1 (1970): 55-67.

CHAPTER 5

CONCLUSIONS

In this thesis, three condition monitoring applications have been introduced which were tried to solve using signal processing, Machine Learning and Deep Learning based algorithms. One application was in the biomedical domain and other two applications were based on fault diagnosis (fault classification and predictive maintenance) in the industrial machines. Various DL algorithms namely VGG16, VGG19, Xception, DenseNet-121, LSTM, ANN,RNN ,GRU and ConvLSTM along with Machine Learning algorithms have been deployed in this chapter for these applications. All the proposed methodology have provided very satisfactory results. Various evaluation metrics such as accuracy, recall, precision, f1-score, AUC score and rmse score have been considered to evaluate the prediction performance.

In chapter 2, audio signals have been used for Parkinson's disease (PD) classification. This chapter introduced various feature extraction techniques which involved Wavelet Transform and Cross-wavelet Transform with VGG16, VGG19, DenseNet-121 and Xception architecture and a comparison study has been done between these proposed methods. Different Machine Learning algorithms along with majority voting classifier have been deployed for the classification.

In chapter 3, fault classification in the bearing has been done using DL algorithms. For this purpose, vibration acceleration signal has been used. The proposed methodology contained feature extraction, feature normalization and then classification using proposed DL models along with weighted average ensemble techniques to improve the classification performance.

In chapter 4, predictive maintenance of the bearing has been discussed which was stated that after how many cycles the machine was going to fail. For this purpose,

features have been extracted from the vibration signal and converted these into time-series sequence. One of the features was taken as condition indicator of the bearing which was analogous with the machine's failure. Then LSTM has been deployed whose performance was measured by the rmse score