A Comparative Study of EMD and FDM for Heart Disease Detection

A project report submitted in partial fulfilment of the requirements for the degree of
Bachelor of Technology

Submitted by

Rana Prafful Rajendra Singh, 194574
Shefali Kanojia, 194564
Priyanshu Singh, 194576
Arsh Guleria, 194514

Under the guidance of
Dr Pushpendra Singh

Major Project-I Report, EC-419
Dual Degree, 4^{th} Year

Department of Electronics and Communication Engineering
National Institute of Technology Hamirpur
Hamirpur, India-177005

Preprint submitted to NIT Hamirpur *August-December, 2022*

Candidate's Declaration

We hereby declare that the research presented in this dissertation titled **"A Comparative Study of EMD and FDM for Heart Disease Detection "** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology and submitted in the Department of Electronics and Communication Engineering of the National Institute of Technology Hamirpur, is an authentic record of our own work carried out during a period from August 2022 to December 2022 under the guidance of **Dr Pushpendra Singh**, Assistant Professor, Department of Electronics and Communication Engineering, National Institute of Technology Hamirpur.

The matter presented in this report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Rana Prafful Rajendra Singh (194574)
Shefali Kanojia (194564)
Priyanshu Singh (194576)
Arsh Guleria (194514)

This is to certify that the above statement made by the candidates is true to the best of my knowledge and belief.

Dr Pushpendra Singh

Date : 15-12-2022 **Assistant Professor, DoECE**

ACKNOWLEDGEMENT

We would like to convey our sincere gratitude to our supervisor, Dr. Pushpendra Singh, Assistant Professor, Department of Electronics and Communication Engineering. Under his guidance, we not only learned about how to approach the problem statement but also gained valuable knowledge regardinghow to conduct research work and how one should always think of finding out a better-optimized solution rather than just coming up with a workable solution. We would also like to extend our heartful thanks for his critical suggestions, helpful criticism, encouragement, and guidance throughout the project. This project would not have been possible without the work and assistance of our group members. The commitment of each team member was essential for the success of this endavour.

Rana Prafful Rajendra Singh (194574)
Shefali Kanojia (194564)
Priyanshu Singh (194576)
Arsh Guleria (194514)

ABSTRACT

Over the years, it has been a perception that Fourier methods are not suitable for the analysis of non-linear, non-stationary time series signals. Empirical Mode Decomposition, developed at NASA, is a widely used method used for the analysis of such signals. It decomposes the signal through the process of sifting. It was also stated that Fourier methods cannot be used for such analysis. Fourier Decomposition method, developed by our supervisor, Dr Pushpendra Singh, is based on Fourier methods and is suitable to be used for the analysis of non-stationary and non-linear signals. In this project we carry out a comparative analysis of such signals using both the methods. We extend this project to classification of heart diseases using Electrocardiogram (ECG) signals. ECG signal is a non-stationary signal and hence useful for our comparative analysis. The Classification is done using Convolutional Neural Networks and their performances are hence compared. We also explore data augmentation, image classification, multivariate time series classification and building a dual input neural network.

Table of Contents

List of Figures

List of Figures

List of Acronyms/Abbreviations

FFT Fast Fourier Transform

STFT Short Time Fourier Transform

EMD Empirical Mode Decomposition

IMF's Intrinsic Mode Functions

FDM Fourier Decomposition Method

FIBFs Fourier Intrinsic Band Function

TFR Time Frequency Representation

ANN Aritificial Neural Network

CNN Convolutional Neural Network

1D CNN One Dimensional Convolutional Neural Network

2D CNN Two Dimensional Convolutional Neural Network

1. Introduction

According to WHO data, Cardiovascular Diseases (CVDs) are the leading cause of deaths globally, taking about 17.9 million lives every year, which is 31% of all global deaths. India accounts for one-fifth of these deaths CVDs are a group of disorders of the heart and blood vessels. Of these deaths, 85% were due to heart attack and stroke. Progressively, a lot of medical studies have been applied so as to effectively treat Heart Diseases. In this project, we build a deep learning model that would, after training, predict a heart disease associated with a given ECG signal.

An Electrocardiogram (ECG) records the electrical signals in the heart [8]. An electrocardiogram is often done in a health care provider's office, a clinic or a hospital room. ECG machines are standard equipment in operating rooms and ambulances. Some personal devices, such as smartwatches, offer ECG monitoring. Show below on the left is the ECG machine and on the right is one period of the ECG signal.

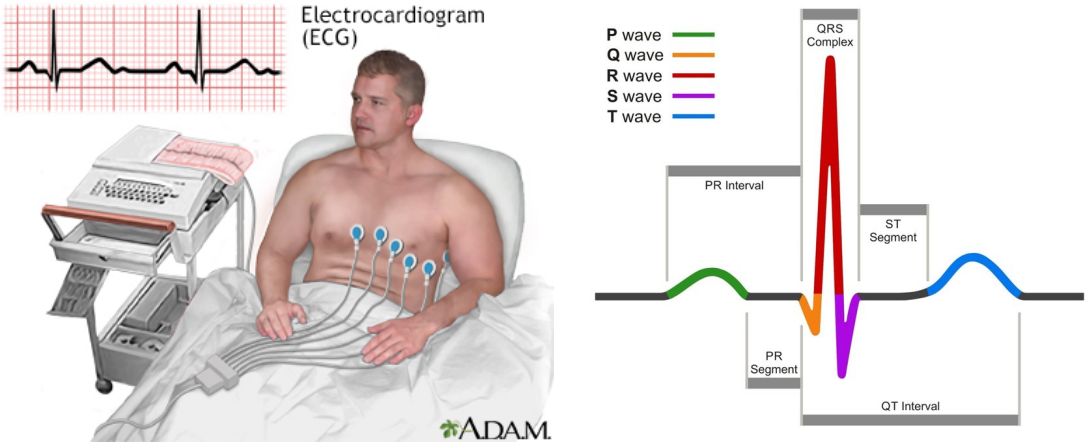


Figure 1: ECG signal of a patient being recorded(left) ECG signal waveform(right)

The frequency range of an ECG signal is 0.05-60 Hz and its dynamic range is 1-10 mV. The ECG signal is characterized by five peaks and valleys represented by the letters P, Q, R, S, T. Sometimes U wave is also present. The performance of ECG analysis is based on the accurate and reliable detection of the QRS complex as well as T and P waves.

1.1. *Objective*

This ECG signal is a Time-series. Hence, we perform Time-series classification with the help of deep learning neural network. We aim to achieve this in three ways, One being a 2D Convolutional Neural Network (2D-CNN), second being a 1D Convolutional Neural Network (1D-CNN) and the third being a hybrid dual input network (2D-CNN & 1D-CNN inputs). We perform the Feature Extraction using two methods, One is the Empirical Mode Decomposition (EMD) and the second is the Fourier Decomposition Method (FDM). We then compare the results obtained using both the methods.

2. Concepts

Here we include the theory and mathematics of the concepts used in the project.

2.1. Heart

Heart is a muscular organ situated between the lungs in the chest cavity. The most important organ of our body has a size of about a clenched fist. The basic task of our heart is to maintain a constant flow of blood throughout the body. Blood in turn carries nutrients, water, oxygen, enzymes etc, with it to cells and tissues which are essential for our sustenance.The Human Heart is divided into four chambers, namely two ventricles and two atria. The ventricles are the chambers that pump blood and the atrium are the chambers that receive blood. Among these, both the right atrium and ventricle make up the right heart, and the left atrium and ventricle make up the left heart. Heart transports blood through arteries that carry blood away from the heart and then it brings it back through the veins. Symptoms of a possible heart problems are chest pain, chest pressure, chest discomfort palpitation, dizziness, pain in neck,jaw ,throat and shortness of breath. Diseases like Coronary artery disease, arrhythmias,Dilated cardiomyopathy, Myocardial infarction, congenital heart defects can be detected with the help of electrocardiogram.

2.2. Electrocardiogram Signals

They are the graphical demonstration of the variation of biopotential versus time. Biopotentials are generated by volume conduction of currents made by collection of electrogenic cells. The human heart contains four chambers that is, Right Atrium, Left Atrium, Right Ventricle and Left Ventricle. Under healthy condition the heartbeat begins at the Right Atrium called Sino Atria (SA) node and a special group of cells send these electrical signals across the heart. This signal travels from the Atria to the Atrio Ventricular (AV) node. The AV node connects to a group of fibers in Ventricles that conducts the electrical signal and transmits the impulse to all parts of the lower chamber, the Ventricles. To ensure that the heart is functioning properly this path of propagation must be traced accurately.

Each heart beat displayed is a sequence of electrical waves characterized by peaks and valleys. ECG mainly provides two kinds of information. One is the duration of the electrical wave passing through the heart and it will decide whether the electrical activity is normal or slow or irregular. Second is the amount of electrical activity passing through the heart muscle that helps to find whether the parts of the heart are too large or overworked. The frequency range of an ECG signal is 0.05-100 Hz and its dynamic range is 1-10 mV. The ECG signal is characterized by five peaks and valleys represented by the letters P, Q, R, S, T. Sometimes U wave is also present. The performance of ECG analysis is based on the accurate and reliable detection of the QRS complex as well as T and P waves. The P-wave represent the activation of the two atria, the upper chambers of the heart, while the QRS complex and T-wave represent the excitation of the ventricles. The P, QRS and T-waves reflect the rhythmic electrical depolarization and repolarization of the myocardium linked with the contractions of the atria and ventricles . The horizontal section of this waveform prior to

the P-wave is termed as the baseline or the isopotential line. The P-wavecorresponds to the depolarization of the atrial musculature. The QRS complex gives the combined result of the repolarization of the atria and depolarization of the ventricles, which occurs almost at same time. The T-wave is the wave of ventricular repolarization, where as the U-wave, if present is normally believed to be the result of after potentials in the ventricular muscle. So the duration amplitude and morphology of the QRS complex is helpful in diagnosing cardiac arrhythmias, conduction abnormalities, ventricular hypertrophy, myocardial infection and other disease states.

The heart beats 60-100 times per minute in the normal state. The slower heart beat state is called bradycardia and a higher rate is called tachycardia. If the ECG signal is not normal then an Arrhythmia is indicated.

Abnormalities	Characteristics
Bradycardia	R-R interval>1s
Tachycardia	R-R interval<0.6s
Hypercalcaemia	QRS interval<0.1s
Dextrocardia	Inverted P-wave
Hyperkalemia	Tall T-wave and absence of Pwave
Sudden cardiac death	Irregular ECG
Sinoatrial block	Complete drop out of a cardiac cycle
Myocardial ischemia	Inverted T-wave

2.3. Time Series

Time series is a series of data points that is ordered in time. Hence, the position of occurrence of a data point in time is important. A time series data is generally formed by sampling data at some interval of time for a long time. Time Series Signals can be further classified as Stationary and non-Stationary Signals which are explained later. Time Series Analysis finds a widespread applications in finance, medicine, astronomy, weather forecasting, business development, etc. The ECG Signal is a non-stationary and non-linear Time-Series signal.

2.4. Stationary and non-Stationary Signals

A signal is said to be stationary if its frequency/spectral and statistical contents are not changing with respect to time. Assume that the signal is represented as X(n) when it satisfies:

1. $E[X(n)] = \mu$
2. $E[\|X(n)\|^2] < \infty]$
3. $r(n1, n2) = E[x(n)x(n+m)] = r(m)$

The signal x(n) is said to be a wide stationary (or generalized stationary) signal. For example, a sine wave at constant amplitude and frequency, is also stationary. White noise is also an example of stationary signal.

On the other hand, signals like speech and time series do not classify as Stationary Signals. Their statistical characteristics tend to change with time along with their spectral components. Hence, classified as non-stationary signals.

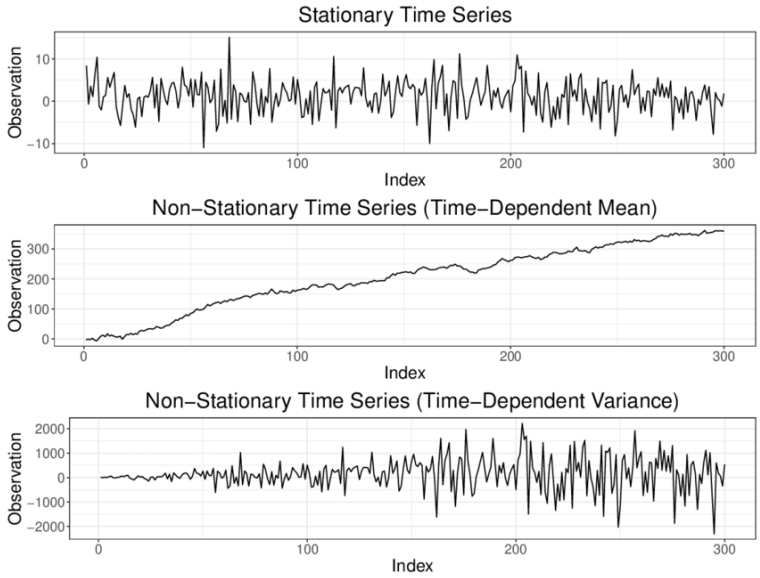


Figure 2: Stationary and Non-Stationary Signals [9]

2.5. Time-Frequency Representation

A Time-Frequency representation (TFR) is a view of the signal, represented over both time and frequency. TFRs are complex valued fields over time and frequency, with the modulus representing the amplitude or energy density and the argument represents phase. TFR representaion in this project is obtained by decomposing the signal into intrinsic band functions and then we apply hilbert's transform to obtain the corresponding TFR.

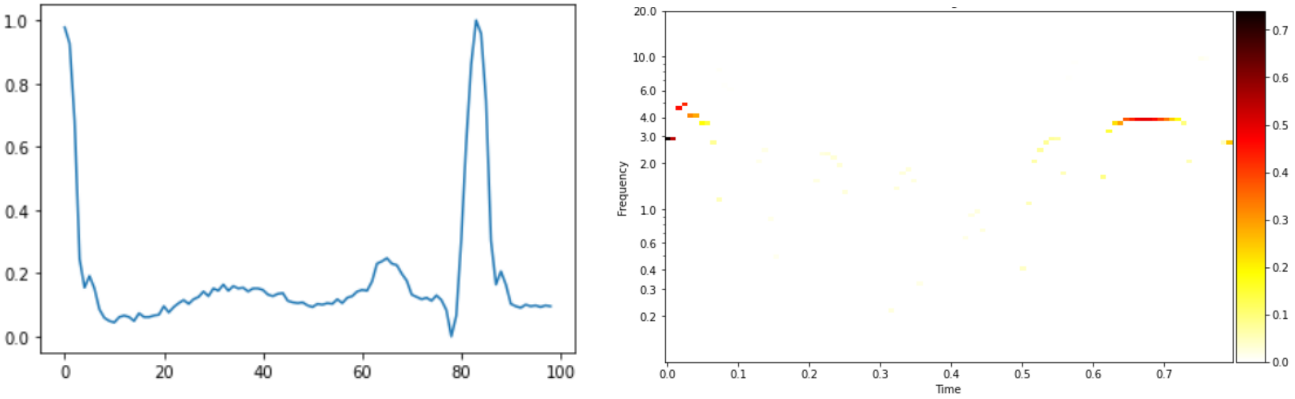


Figure 3: Time-Frequency Representation of corresponding ECG Signal

Here the darkness of colour represent the amplitude. Darker the colour, more the amplitude.

2.6. Signal Analysis Methods

The ECG signal analysis used to be carried out in Time Domain. But it was not sufficient to study all the characteristics of ECG signal. We need to extract from the ECG Signal. To be able to extract important information from this signal, which is in time domain, we have

to transform it into frequency-time-amplitude domain. To have a better understanding of these signals, various signal processing methods are used like Fast Fourier Transform, Short-Time Fourier Transform, Wavelet Transform, etc. We would now have an overview of these methods

i) Fast Fourier Transform: It is a method that transforms time domain signal into frequency domain signal. This is a fundamental transform used in digital signal processing and has various applications.

$$X(w) = \sum_{n=0}^{N-1} xe^{-2\pi nki/N} \tag{1}$$

However FFT fail to provide information regarding the accurate location of frequency component in time.

ii) Short-Time Fourier Transform: To overcome the shortcomings of FFT, a windowed fourier transform was introduced. also known as Gabor transform. Short-Time Fourier Transform (STFT) has both time and frequency information. STFT spectrogram is a simple and fast technique in comparison to other time-frequency analysis. THe technique slices the waveform of interest into a number of short segment and analyse each segment using standard fourier transform.A window function is applied to a segment of data, efficiently isolating that segment from the overall waveform, and Fourier transform is applied to that segment.

$$X(\tau, f) = \int_{-\tau/2}^{\tau/2} x(t)w(t - \tau e^{-2i\pi ft})dt. \tag{2}$$

The issue with STFT is that it's time frequency precision is not optimal.

iii) Wavelet Transform: It has a multiresolution property which gives both time and frequency information through variable window size. A wavelet is a small wave which has energy concentrated in time and provides a tool for analysis of transient, non-stationary or time-varying signals.The Wavelet Transform is a time-scale representation that has been used effectively in a variety of applications, in particular signal compression. It is a linear process that decomposes the signal into a number of scales associated with frequency components and analyzes each scale with a certain resolution.

Although these approaches have many useful applications, however, the analysis of non-stationary signals are not well presented by these methods. The recently proposed empirical mode decomposition (EMD) provides a general method for examining the Time Frequency Distribution.

2.7. Empirical Mode Decomposition

it is an adaptive and efficient decomposition method capable of decomposing any complex signal into finite intrinsic mode functions [1]. The Data itself dictates the decomposition.

It is suitable for processing non-linear, non-stationary signal analysis. The decomposition is accomplished using empirical bases termed Intrinsic Mode Functions (IMF's) that satisfy the following properties:
i) For the whole dataset, the number of extrema and number of zero crossings must either equal or differ by atmost one.
ii) At any point, the mean value of the envelope defined by the local maxima and envelope defined by local minima is zero.
Contrary to almost all previous signal analysis methods, this new method is intuitive, direct and adaptive.

$$x(t) = \sum_{i=1}^{l} y_i(t) + r_l(t) = \sum_{i=1}^{l+1} y_i(t) \tag{3}$$

All the IMFs must satisfy two basic conditions:
(i) in the complete duration of time series, the number of extrema (i.e. maxima and minima) and the number of zero crossings are equal or differ at most by one.
(ii) At any point of time in the complete duration of time series, the average of the upper and lower envelopes, obtained by the interpolation of local maxima and the local minima, is zero.
The first condition ensures that IMFs are narrow band signals and the second condition is necessary to ensure that the IF does not fluctuate excessively because of asymmetry of waveforms.
To extract an imfs, we first create two envelopes. One envelope touches all the local maxima while the lower envelope touches all the local minima. The average of this signal is determined and then a signal is obtained which is the difference of the signal and the average of the envelope. We then see if the resultant signal satisfies the condition of being an imf. if yes then same process is applied on (signal-imf) and rest imfs are extracted. if no then (signal-resultant) is iterated with the above process to obtain the first imf. This process is called sifting.

2.8. Fourier Decomposition Method

A class of functions, termed as the Fourier intrinsic band functions (FIBFs) [2], belonging to C [a, b], here with the following formal definition.Definition 1: The Fourier intrinsic band functions (FIBFs), $y_i(t) \in$ C-[a, b], are functions that satisfy the following conditions:
i) The FIBFs are zero mean functions, i.e. $\int_a^b y_i(t)dt$ =0.
ii) The FIBFs are orthogonal functions, i.e.$\int_a^b y_i(t)y_j(t)dt = 0$,for i $\neq j$.

Thus, the AFIBFs are monocomponent signals and, physically, the IF has meaning only for monocomponent signals, i.e., signal has only one frequency or a narrow range of frequencies varying as a function of time. Thus, the FIBF is sum of zero mean sinusoidal functions of consecutive frequency band.The main objective of this study is to obtain unique representation of multicomponent signal as a sum of constant and monocomponent signals, i.e. signals

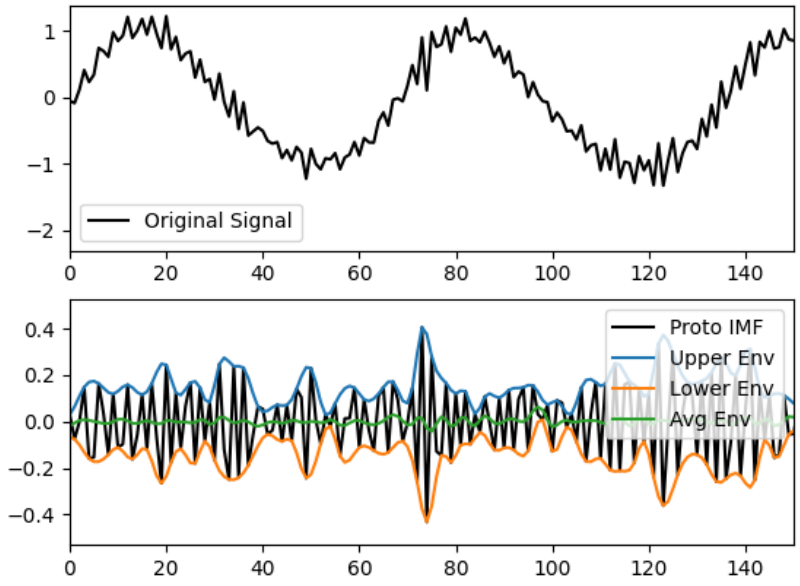


Figure 4: Sifting process [7]

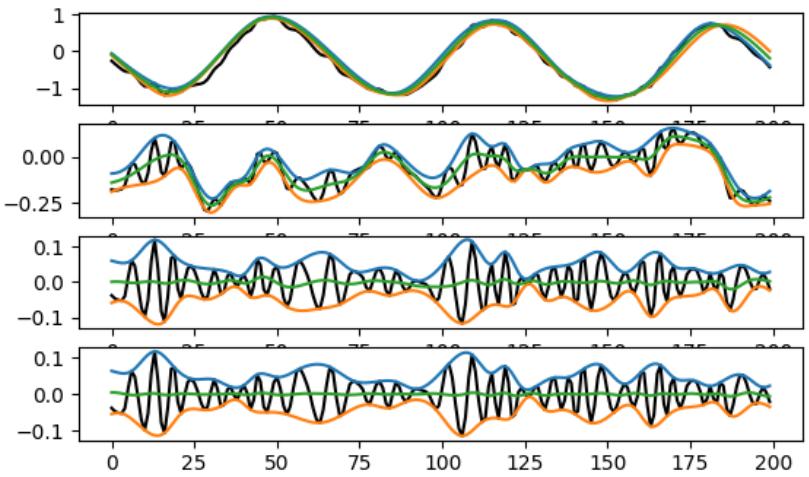


Figure 5: Complete Sifting of the sample

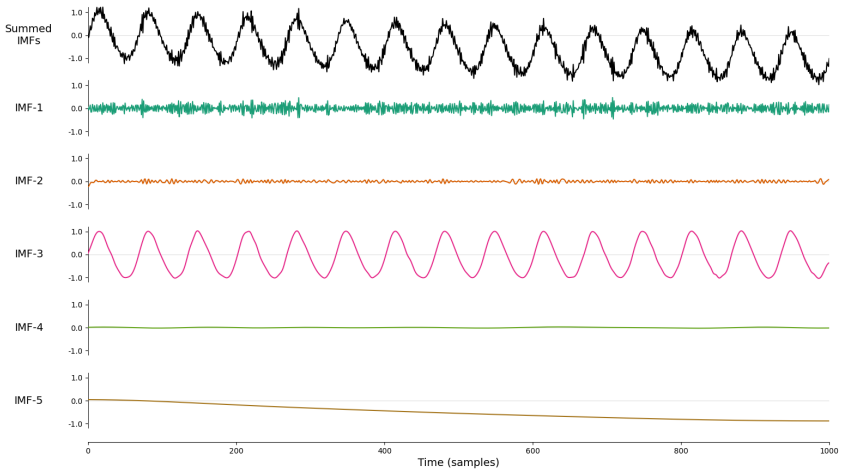


Figure 6: Intrinsic Mode Functions of a Signal (Sifting), First sample is the signal while others are the IMF

which can be represented by the following model.

$$x(t) = \sum_{i=1}^{M} y_i(t) + n(t) \tag{4}$$

where n(t) is a noise representing any residue (constant or trend) components and the y(t) are M single component nonstationary signals, would be the FIBFs, defined above. Below is the FIBFs of a sample signal from MIT-BIH Dataset.

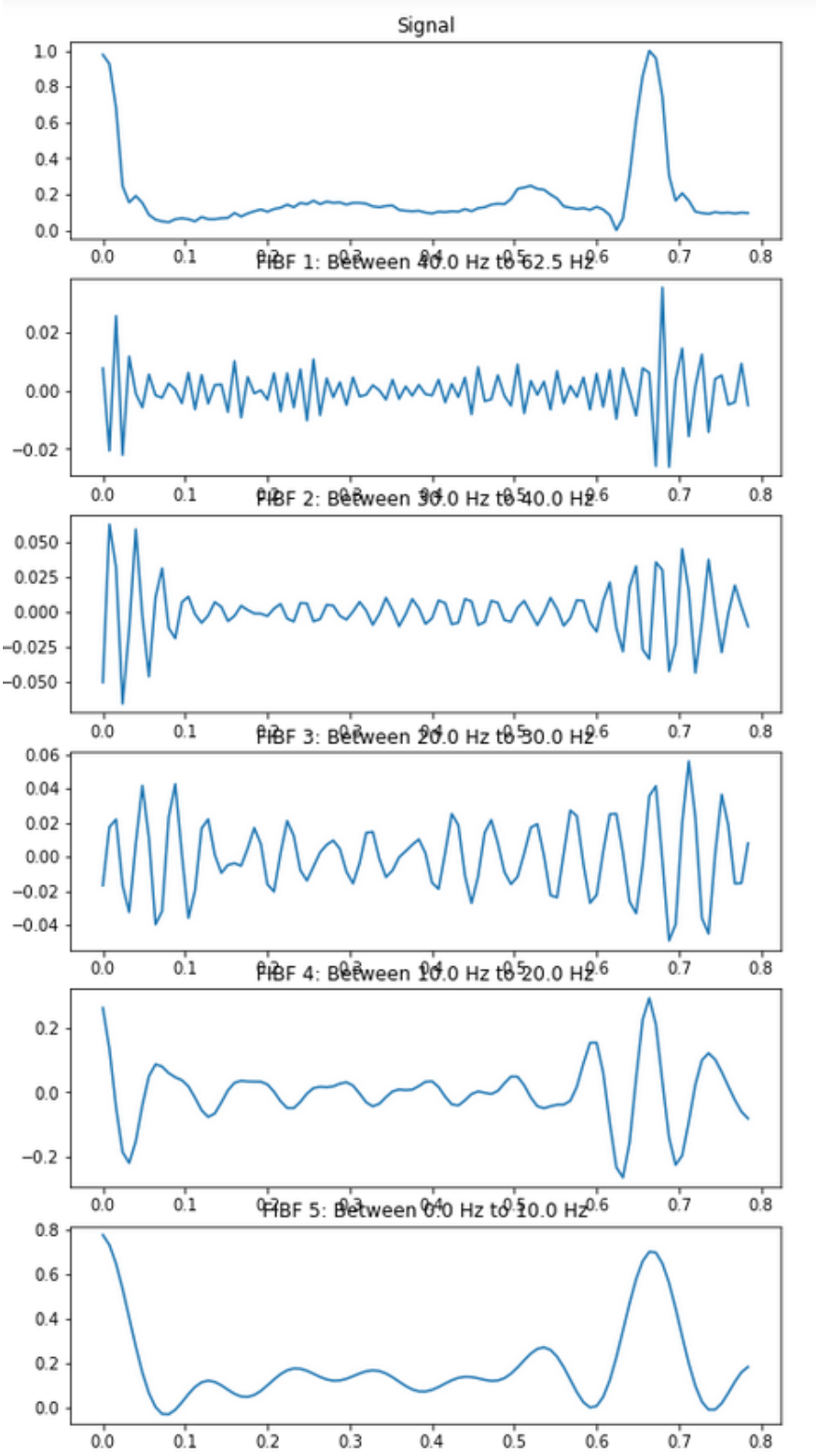


Figure 7: Fourier Intrinsic Band Functions of a given signal. First sample corresponds to the signal and all others are the FIBFs of that signal.

2.9. *Classification*

Classification refers to grouping of similar objects together. In our case, we attempt to group the ECG signals having the same class of disease. In machine Learning, there are various algorithms used for classification of the data and they include:
Logistic Regression Support Vector Machines K-Nearest Neighbours Kernel SVM Naïve Bayes Decision Tree Classification Random Forest Classification Neural Network Classification
Our data is a time series data. We attempt to do a image classification as well as multivariate time series classification. So we choose Neural Network Classification. In particular, Convolutional Neural networks (CNN).

2.10. *Neural Network*

A Neural Network is a set of layers, contained of nodes, where nodes of a layer are connected to the nodes of the layer to its right. These links are associated with some weights and biases. Initially these nodes are random. During training, the neural network uses the concept of back propagation and gradient descent to set these weights and biases such that this combination of them provides with the most accurate results.

2.11. *4-folds cross-validation*

Conventionally in hold-out validation, we split data into training and validation set in the ratio of 4:1. However, to be better able to judge the performance of our model we use k-fold cross validation and take k=4. In this approach, we shuffle and split our data into 4 partitions of equal size. For each part i, train the model on the remaining 3 parts and evaluate it on part i. The final score is the average of the 4 validation scores obtained.

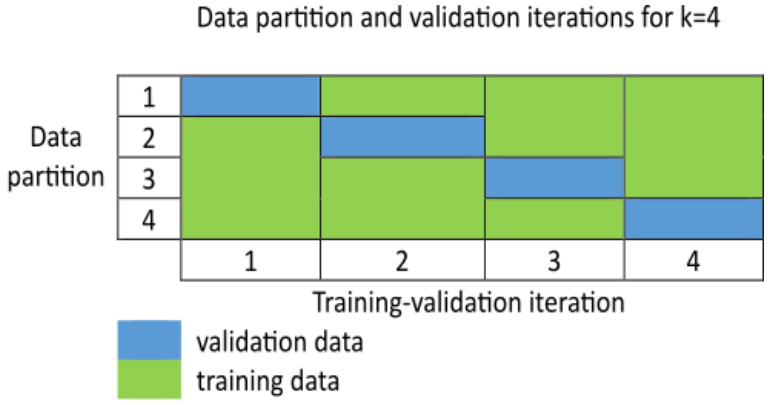


Figure 8: 4-folds cross-validation schematic [6]

3. The Dataset

To carry out the analysis , we need samples of ECG signals with various classes of diseases well represented. To train a Deep Learning model, more the samples the better it is. The ECG Heartbeat dataset used is the MIT-BIH Arrythmia Dataset [4]. This Dataset was the first generally available set of standard test material for evaluation of arrhythmia detectors, and it has been used for that purpose as well as for basic research into cardiac dynamics at about 500 sites worldwide since 1980.

3.1. Classes

The MIT-BIH Dataset contains the ECG signals corresponding to five different categories referred as classes. It contains 109,446 samples of 5 different label Classes: ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4] where these correspond to N: Normal beat, S: Supraventricular premature beat, V: Premature ventricular contraction, F: Fusion of ventricular and normal beat, Q: Unclassifiable beat.

3.2. Samples Overview

The Samples are obtained at a sample rate of 125Hz. Each sample is zero padded so as to make them of the same size of 187. The 188th sample point corresponds to the class of disease associated with the sample concerned. The samples are divided into train and test dataset containing 87553 and 21893 samples.

3.3. Problems and their Solutions

Though the number of samples in the training dataset is 87556 ,the distribution of labels are very uneven. The frequencies of the five label classes in the data-set are [72471, 2223, 5788, 641, 6431]. 82.77% represent label 0 disease while only 0.73% represent label 3 disease. This uneven distribution needs to be eliminated. Also for the purpose of analysis, we need to remove the zero padding from each samples before further analysis. We encounter these problems as follows.

3.3.1. Removing Zero Padding

Zero Paddings are removed with the help of the following algorithm. After slicing away the classes array from the sample, each sample is looped from the last sample to first. As soon as we encounter a non-zero value, we stop iterating. The un-padded signal is the one from 0th index upto this point. As we shall see, padded signals produce unwanted signals in the intrinsic functions

```
def remove_padding(data):
end = len(data)-1 #pointer from the last data point
while data[end]==0: #continue if current data is 0.
    end=end-1
return data[:end] # return data up until break point,
```

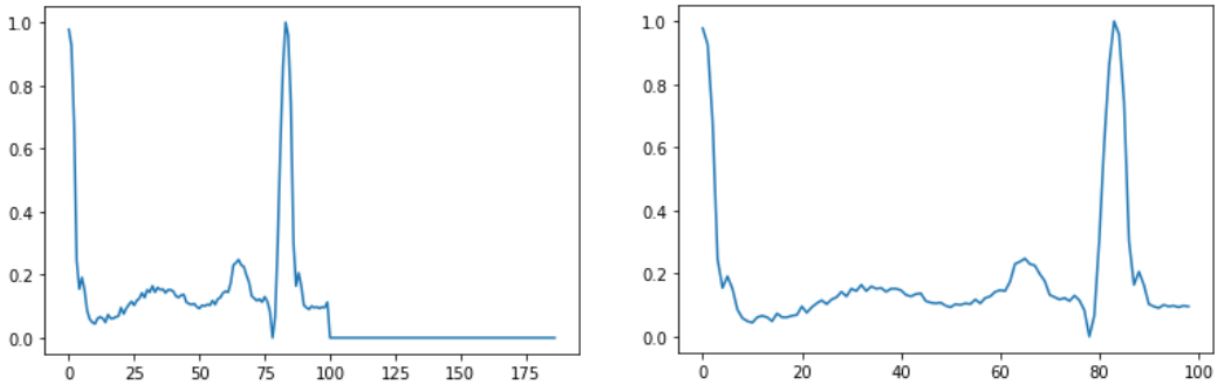


Figure 9: Sample ECG Data, with(left) and without(right) padding.

3.3.2. Data Augmentation

One way of having equal representation of classes in the dataset is to include that many samples from each classes as the number of samples of rarest class in the dataser. In our case, the class 3 disease had 641 samples. So we simply include 500 test and 120 test samples from each class of dataset. Since, 2500 train samples are very low number of samples, overfitting is expected and this will be shown later in the report.

Now, a proposed way to increase the total number of samples, along with even representation of classes is by applying Data Augmentation on the data segregated as above. Data Augmentation is a technique in which we generate new and unique samples from already existing samples. These new samples differ from the existing samples that they are derived from. We make use of the technique of the **Phase-Shift Method**. This method shifts the phase of the frequency component of the signal by the value α we specify. If this shift id$\pi/2$ then it becomes the hilberts transform of the signal. Note that the energy content of the signal still remains the same beause it's only the phase that is shifted.

$$X(t, \alpha) = x(t)cos\alpha + x^h(t)sin\alpha \tag{5}$$

Here $x^h(t)$ represents the hilberts transform of x(t). For different values of α we get many new sample X(t) from existing sample x(t). We use this to generate new under-represented labels so that they reach a count of about 6000 samples. Hence, we are able to scale up our training samples from 2500 to about 30,000 using Phase-Shift method for Data Augmentation.

4. Feature Extraction and Disease Classification

The data points in time is a feature of a particular dataset sample. Hence a one dimensional representation of a signal. But, it is better to have a multiple feature reprentation of a sample. For Example, to tell if a fruit is tasty or not, size could be one of the feature based on which we can make our classification. We can add another feature, like the colour of the fruit, and make the classification. In this case we have two features based on which we make our classification. Without any loss of generality, two feature classification is better than a single feature classification because we have multiple metric to judge a sample. Hence, in our dataset, we obtain a two-dimensional representation of the signal, the two dimensions being time and frequency. Also in the process we obtain the multivariate representation of the sample, that is, a single sample gets represented by a set of time series. The Empirical Mode Decomposition (EMD) and Fourier Decomposition Method (FDM) are the methods by which we can obtain multivariate time series representation of a particular sample. Applying the Hilberts transform to this multivariate time series representation gives the TFR of that sample.

After we are able to obtain the desired representation of our sample, we train our neural network. We have used Convolutional Neural Networks. We carry out three types of classifications,

- Image Classification of TFRs, using 2D Convolutional Neural Networks.
- Multivariate Time Series Classification, using 1D Convolutional Neural Networks.
- Dual Input Classification with TFRs and Multivariate Time Series, Using concatenated 2D Convolutional base and 1D Convolutional base.

Proceeding ahead, we show how we obtained the TFRs and Multivariate Time Series Representation of the dataset samples in the feature extraction section below. Then we define our model architecture and the performances that we obtained using them with the particular feature extraction method applied.

4.1. Feature Extraction

4.1.1. Using Empirical Mode Decomposition

The Concept of EMD has already been explained. We use the method to obtain a set of time series signals called intrinsic mode functions (IMFs). This set is the Multivariate time series representation of the sample. To implement this method we use the EMD library [**T**]his can be done by the use of the function **sift** and the process is called sifting. So to get the imfs of a sample, we simply use :

```
#Generating the IMFs for all the Samples
imfs = [] # list of imfs of all samples
for i in range(length):
    imfs.append(emd.sift.sift(samples[i])) #sift and add to the list
emd.plotting.plot_imfs(imfs[0]) #Plotted a sample
```

This plots the imfs of the 0th sample.

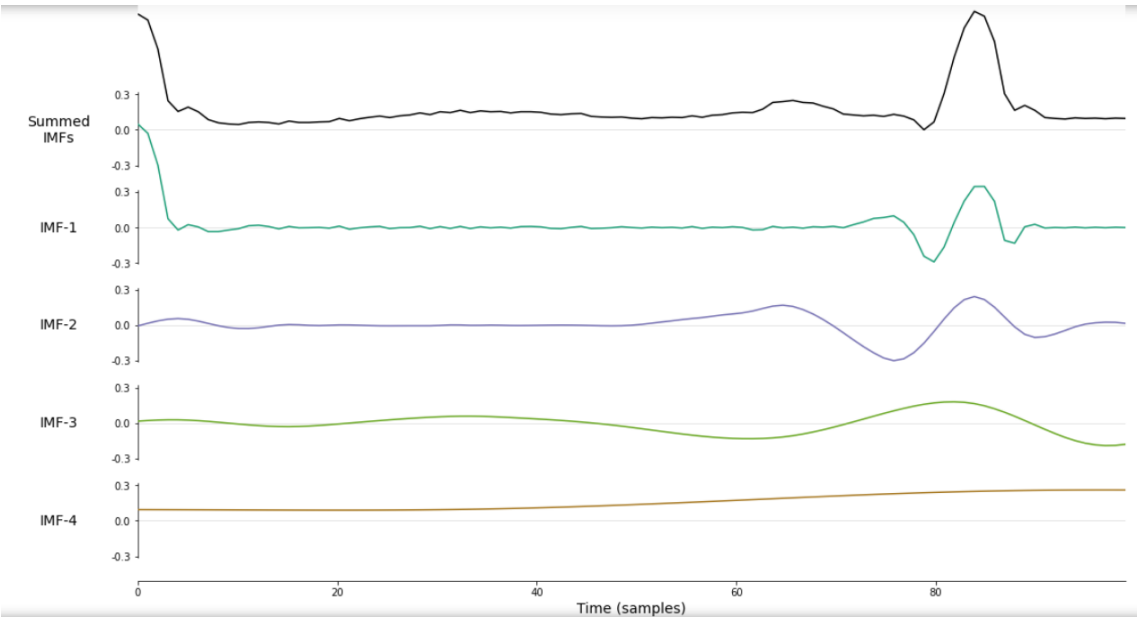


Figure 10: Obtained IMFs of sample[0]

The Summed IMF is the original sample and all it's IMFs are labelled. It simply means, the superposition of all the IMFs form the sample. This is how we obtain our time series representation of the Signal.
Without removal of padding we get the signal as shown in the next figure. Clearly, we get disturbances in the IMFs since padded zeros are not the part of the signal. Now, we pad each imf with zero again to make their sizes equal. This set will form our Multivariate Time Series Data which is fed into 1D CNN.
Now we obtain the TFR using hilbets Transform obtained as

```
    IP, IF, IA =
        emd.spectra.frequency_transform(imfs[i],sample_rate,'hilbert')
        #setting parameters
    freq_range = (0.1,10,80,'log')
    f, hht = emd.spectra.hilberthuang(IF,IA,freq_range,sum_time=False)
        #Producing Hilberts Transform
```

IP, IF, IA corresponds to instantaneous phase, frequency and amplitude respectively.

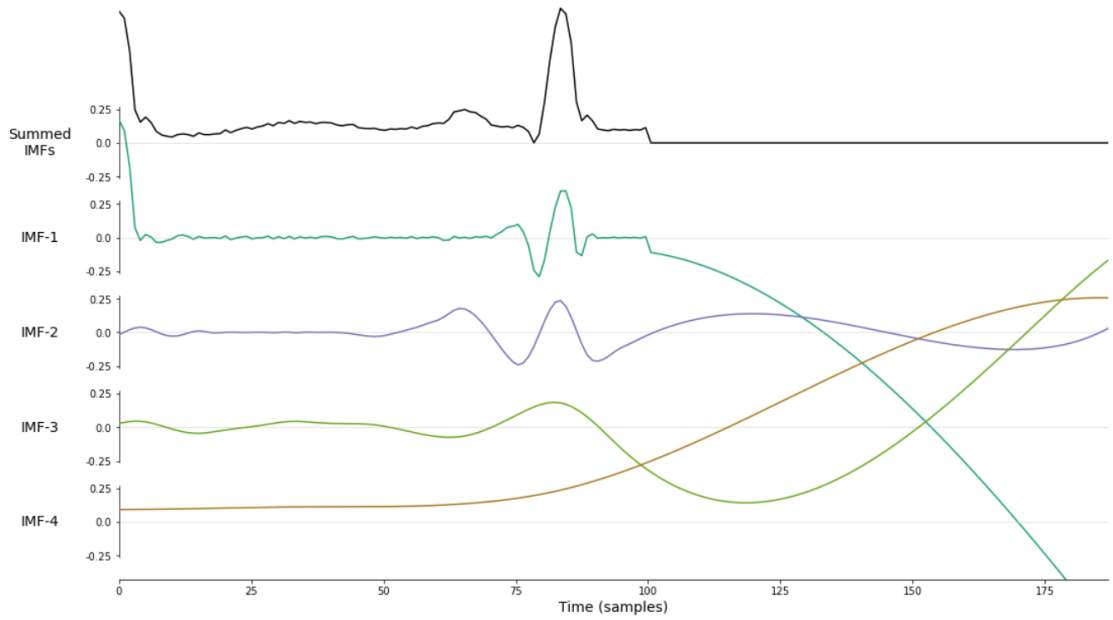


Figure 11: IMF of padded sample[0]

```
# Block to plot Hilbert Transform of the obtained IMFs
sample_rate = 125 #sample rate
fig = plt.figure(figsize=(10, 6))
time_vect = np.linspace(0,maximum/sample_rate,186) # parameters of the plot
emd.plotting.plot_hilberthuang(hht[i], time_vect,f[i],time_lims=(0, 2),
    freq_lims=(0.1, 30),fig=fig, log_y=True) #plot the TFR
```

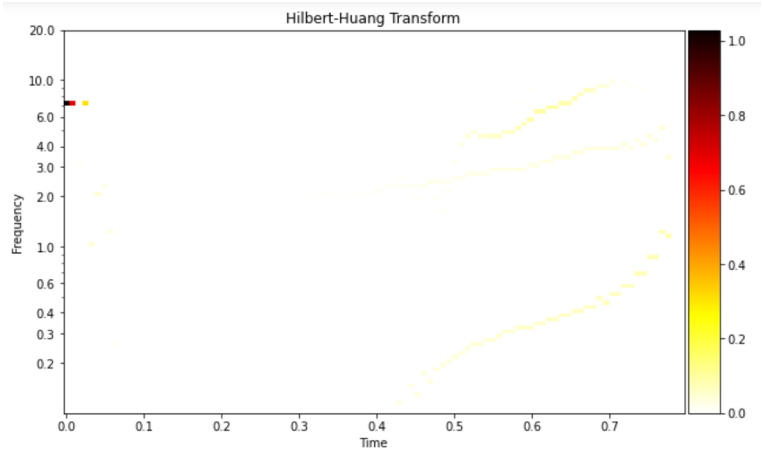


Figure 12: TFR obtained Using EMD

4.1.2. Using Fourier Decomposition Method

The Concepts of the FDM have been discussed earlier. We Decompose the Signal into Fourier Intrinsic Band Functions (FIBFs) which contains the part of the signal corresponding to some range of frequencies. We make use of the FDM library [.] Use of fdm() method provides us with the FIBFs. We then use the hilbert transform method from EMD library similarly.

```
fibfs[i] =
    fdm(sample[0],fs,fc=np.array([0,10,20,30,40]),filter_type='dct',plot_subbands=True)
    #obtain and plot FIBFs
IP, IF, IA = emd.spectra.frequency_transform(fibfs[i],sample_rate,'hilbert')
freq_range = (0.1,10,80,'log')
f, hht = emd.spectra.hilberthuang(IF,IA,freq_range,sum_time=False)
```

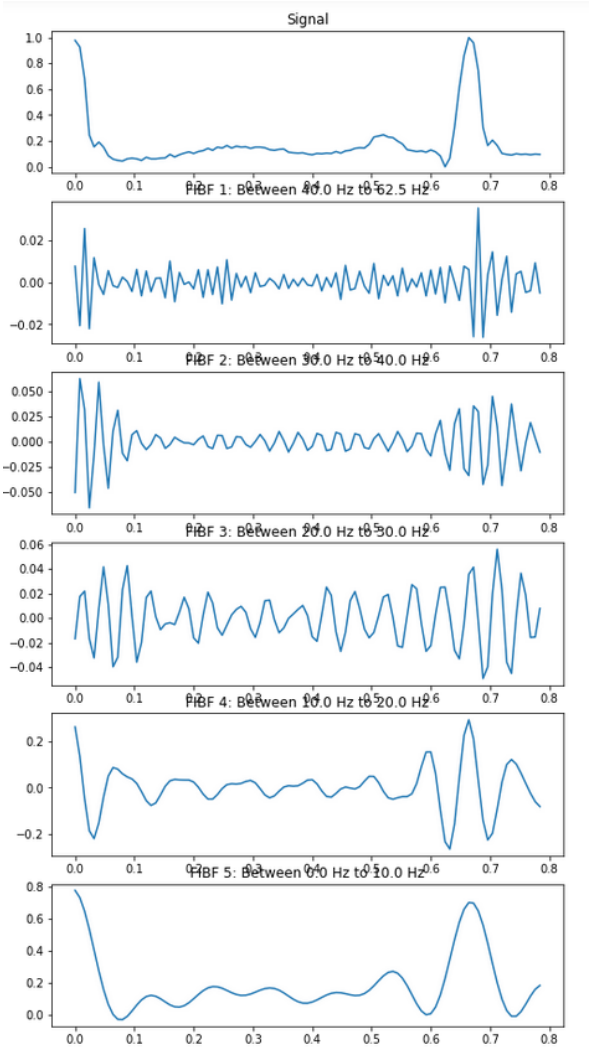


Figure 13: FIBFs of sample[0]

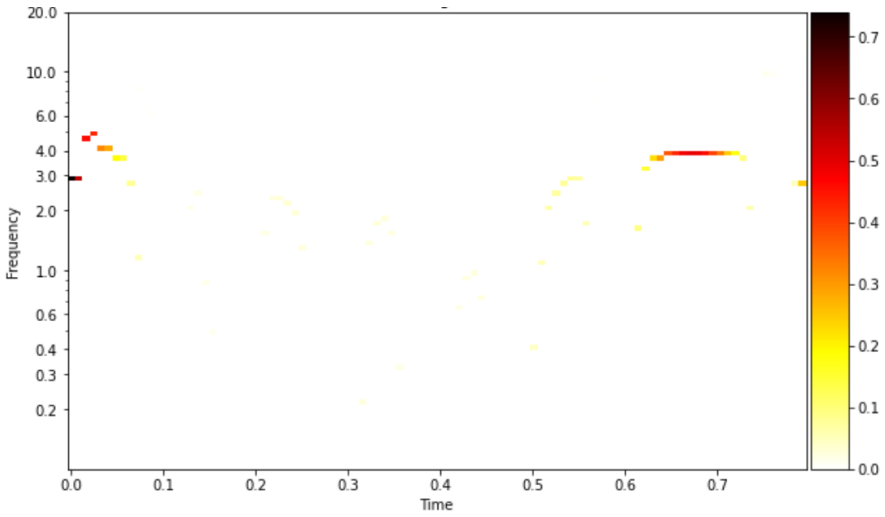


Figure 14: TFR Obtained Using FDM

4.1.3. Comparing TFR's

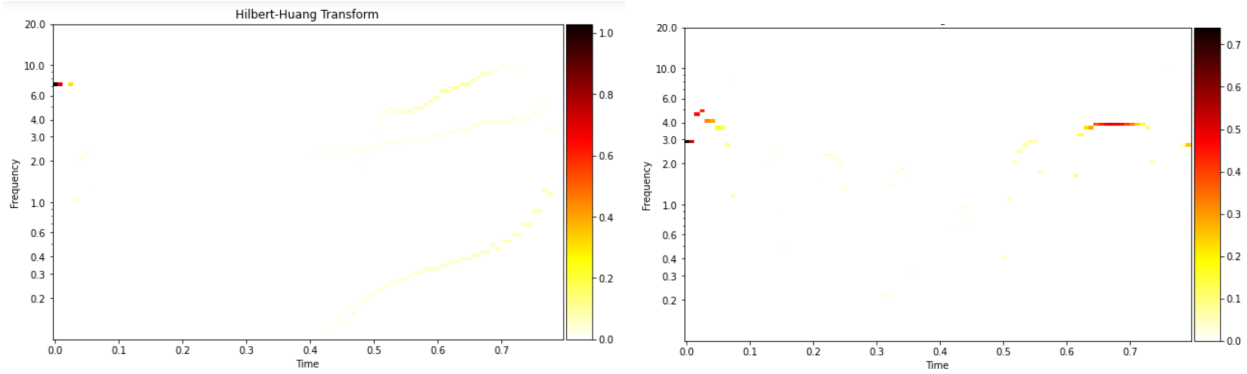


Figure 15: TFR using EMD vs TFR using FDM of sample[0]

We can see that the TFR obtained using FDM has better spectra than that obtained using EMD.

4.2. Deep Learning Classification

We earlier said that due to under-representation of the classes in the dataset, we restricted our dataset to 500 samples per class (+120 for validation), 2500 total samples (+600 for validation). Later, we implemented data Augmentation and increased the dataset to 6000 samples per class, 30000 samples total. We use these samples to train the model. We use 80:20 criteria for training:Validation, We have another dataset of about 21000 samples that we use for testing.

4.2.1. 2D Convolutional Neural Network

Here we carry out TFR Image Classification Using 2D CNN. The 2D CNN model is shown.

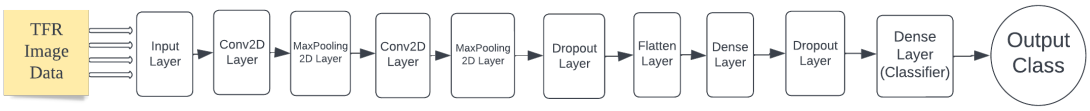


Fig. 2D CNN Classifier Model

```
from keras import layers,models
model = models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation = 'relu',
    input_shape=(80,186,1)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Dropout(0.5))
model.add(layers.Flatten())
model.add(layers.Dense(64,activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(5, activation='softmax'))
model.compile(optimizer =
    'RMSprop',loss='categorical_crossentropy',metrics=['accuracy'])
```

With limited 6000 samples we obtain the following performance:

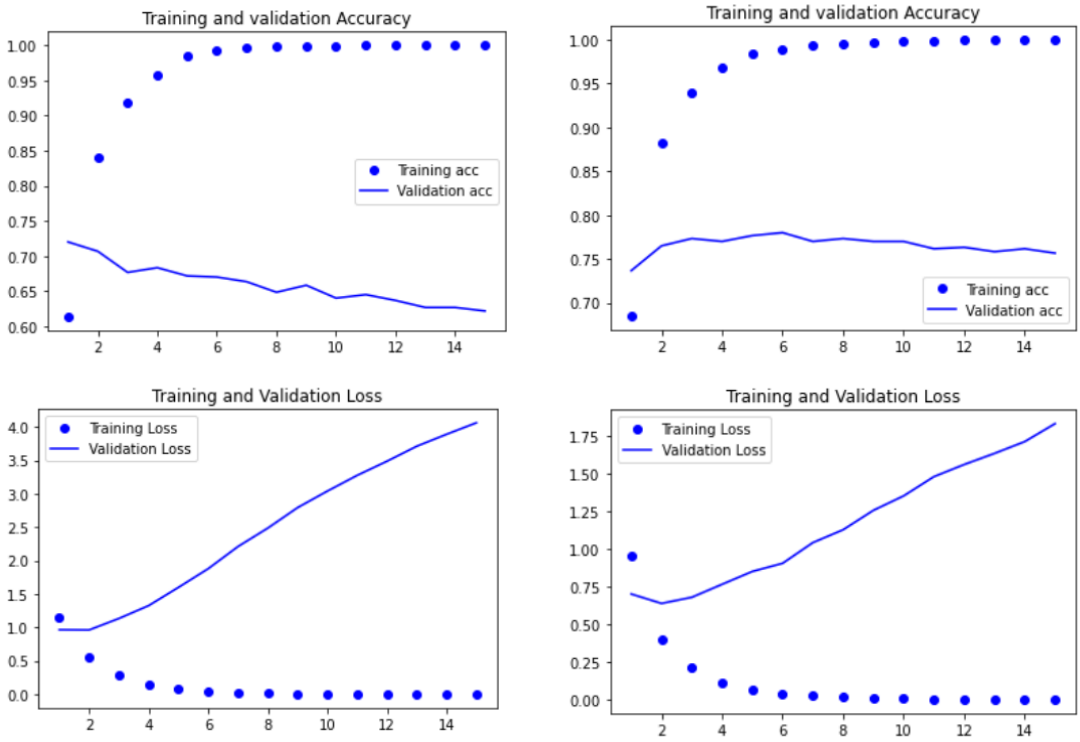


Figure 16: EMD vs FDM, performance using 2D CNN with limited samples

Hence, we need more number of samples and so we implement data augmentation. The following performance is obtained:

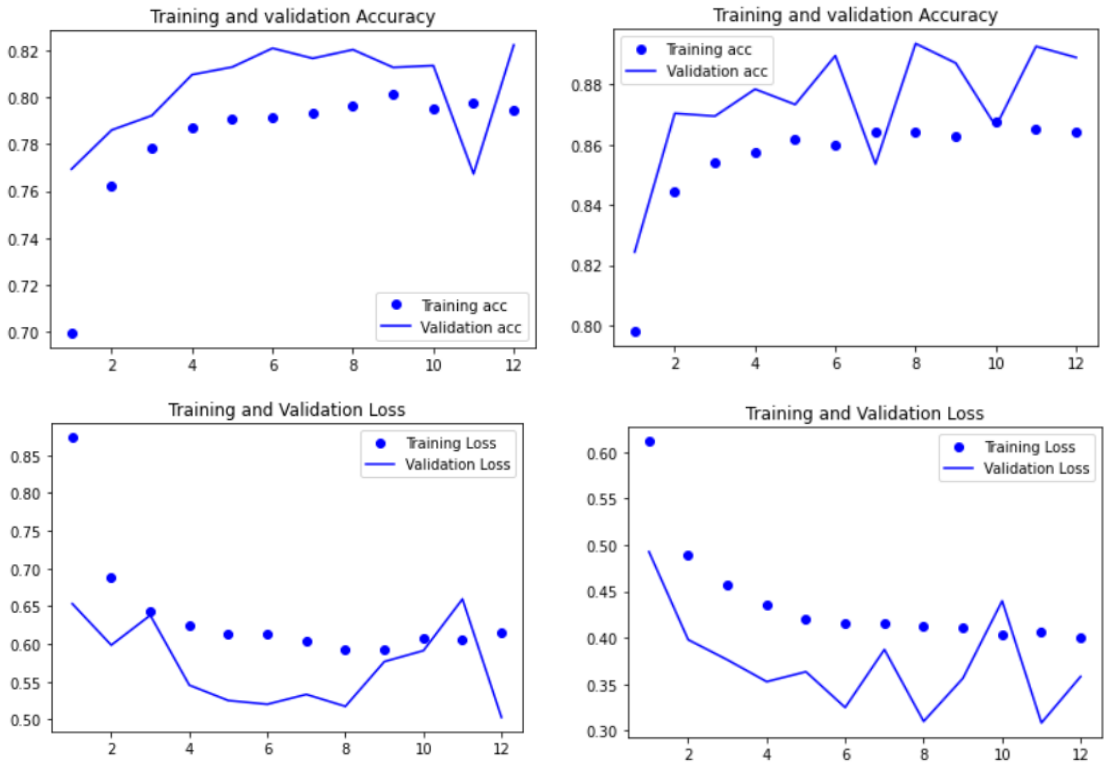


Figure 17: EMD vs FDM, performance using 2D CNN with Data Augmentation

- **With EMD:** The 4-fold cross validation scores obtained were (in the form of [loss, accuracy]) [[0.5551, 0.8043], [0.5414, 0.8183], [0.5054, 0.8210], [0.5867, 0.8099]] with the average of [0.5472, 0.8156]. Hence, the average validation accuracy of 81.56% is obtained.
  On test data, the performance obtained were [0.7953, 0.7475], hence 74.75% test accuracy.

- **With FDM:** The 4-fold cross validation scores obtained were (in the form of [loss, accuracy]) [[0.3623, 0.8821], [0.4316, 0.8724], [0.3021, 0.8972], [0.3045, 0.8952]] with the average of [0.3501, 0.8867]. Hence, the average validation accuracy of 88.67% is obtained.
  On test data, the performance obtained were [0.4361, 0.8611], hence, 86.67% test accuracy.

- **Comparison:** It can clearly be seen that FDM provides a far better performance than EMD with TFR classification. Not only does the model learn better with FDM, but has a significantly better performance on test data than EMD.

4.2.2. 1D Convolutional Neural Network

Here we carry out Multivariate Time Series Classification Using 1D CNN. The 1D CNN model is shown.

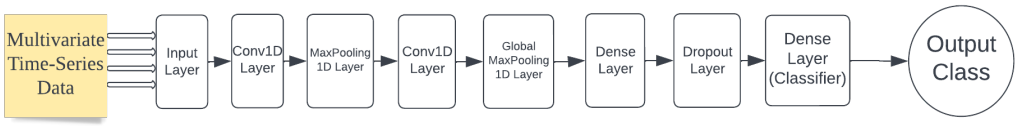


Fig. 1D CNN Classifier Model

```
from keras import layers,models
input_seq = layers.Input(shape = (186,6))
conv1 = layers.Conv1D(24,7,padding="same", activation="tanh")(input_seq)
pool1 = layers.MaxPooling1D(5)(conv1)
conv2 = layers.Conv1D(24,7,padding="same", activation="tanh")(pool1)
processed = layers.GlobalMaxPooling1D()(conv2)
compressed = layers.Dense(50, activation="relu")(processed)
compressed = layers.Dropout(0.3)(compressed)
out = layers.Dense(5, activation='softmax')(compressed)
model = models.Model(inputs=input_seq, outputs=out)
model.compile(optimizer =
    'RMSprop',loss='kullback_leibler_divergence',metrics=['accuracy'])
```

With 2D CNN we saw, that the model was subjected to high overfitting and hence gave bad validation score. Here, we use limited samples with 1D CNN and obtain the following performance.

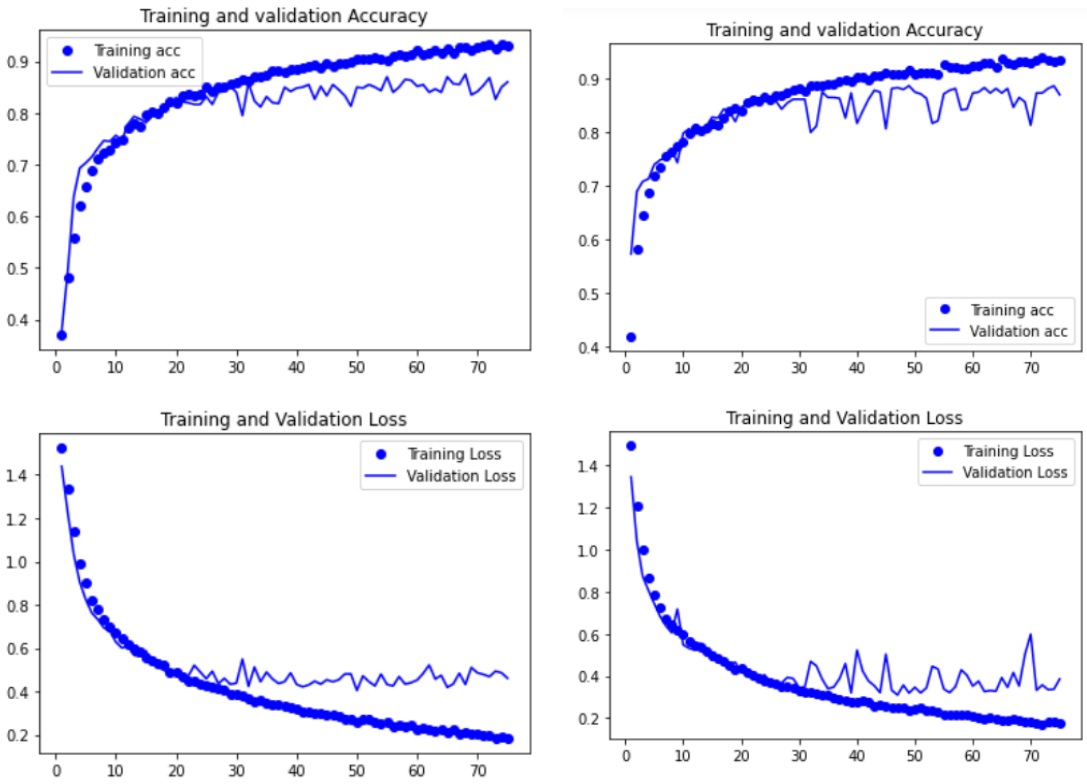
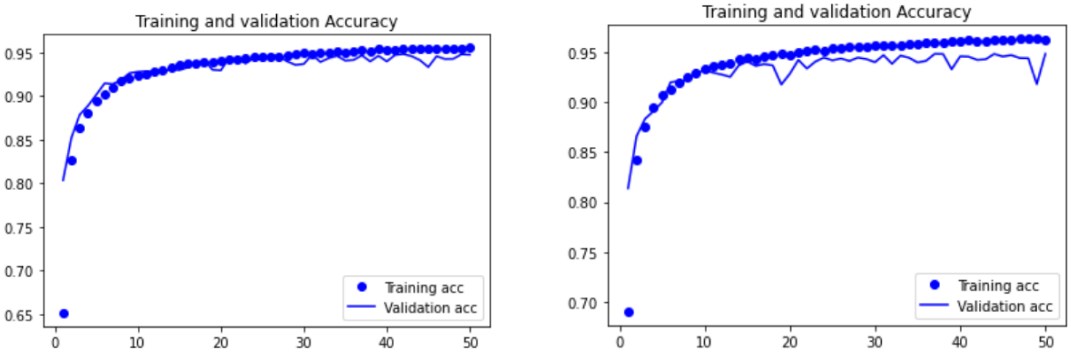


Figure 18: EMD vs FDM, performance using 1D CNN with limited samples

- **With EMD:** The test scores obtained are (in the form of [loss, accuracy]) [0.5680850148200989, 0.8254156708717346], hence 82.54% of test accuracy is obtained.
- **With FDM:** The test scores obtained are (in the form of [loss, accuracy]) [0.43979066610336304, 0.8410835266113281], hence 84.11% of test accuracy is obtained.
- **Observe** that 1D CNN is able to learn from limited samples and provide us with really good test accuracy.

With Data Augmentation, following performance is obtained:



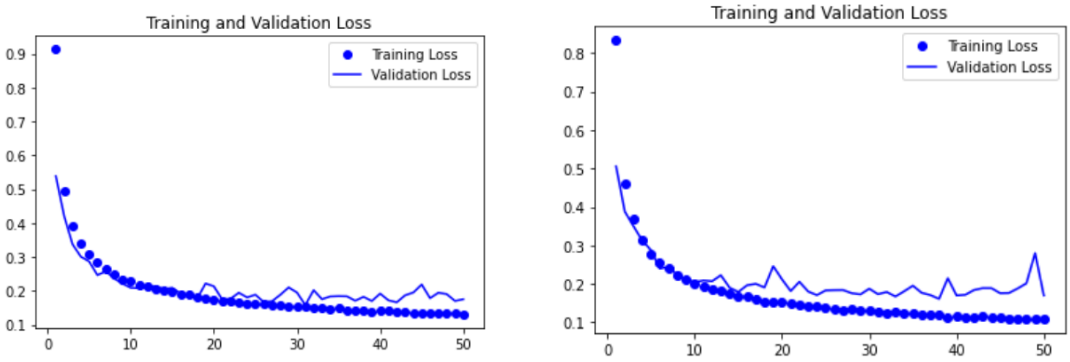


Figure 19: EMD vs FDM, performance using 1D CNN with Data Augmentation

- **With EMD:** The 4-fold cross validation scores obtained were (in the form of [loss, accuracy]) [[0.1754, 0.9459], [0.2296, 0.9398], [0.19045, 0.9419], [0.1702, 0.9485]] with the average of [0.1914, 0.9440]. Hence, the average validation accuracy of 94.4% is obtained.
  On test data, the performance obtained were [0.2696, 0.9306], hence test accuracy of 93.06%.

- **With FDM:** The 4-fold cross validation scores obtained were (in the form of [loss, accuracy]) [[0.1555, 0.9545], [0.1526, 0.9551], [0.1481, 0.9541], [0.1600, 0.9539]] with the average of [0.1540, 0.9544]. Hence, the average validation accuracy of 95.44% is obtained.
  On test data, the performance obtained were [0.1629, 0.9564], hence test accuracy of 95.64%.

- **Comparison:** Here, the performance is competitive, with FDM providing slightly better performance than EMD. Also, significant improvement is obtained in performance when compared with 2D CNN classification for both the methods.

4.2.3. Dual Input Neural Network

Here we carry out classification using both TFRs and multivariate time series data [3]. We have two bases, 2D CNN and 1D CNN respectively. 2D CNN bases takes in TFR input while 1D CNN takes in 1D CNN input. The outputs of the bases are then concatenated together and fed into dense classifier network that outputs the classified class [5]. The model is described below:

```
from keras import layers,models
def get_1dconv():
    input_seq = layers.Input(shape = (186,6))
    conv1 = layers.Conv1D(24,7,padding="same", activation="relu")(input_seq)
    pool1 = layers.MaxPooling1D(5)(conv1)
    conv2 = layers.Conv1D(24,7,padding="same", activation="relu")(pool1)
```

```
    processed = layers.GlobalMaxPooling1D()(conv2)
    compressed = layers.Dense(50, activation="relu")(processed)
    compressed = layers.Dropout(0.3)(compressed)
    model = models.Model(inputs=input_seq,outputs=compressed)
return model
#-----------------------------------
def get_2dconv():
    model = models.Sequential()
    model.add(layers.Conv2D(32,(3,3),activation = 'relu',
        input_shape=(80,186,1)))
    model.add(layers.MaxPooling2D((2,2)))
    model.add(layers.Conv2D(64,(3,3),activation='relu'))
    model.add(layers.MaxPooling2D((2,2)))
    model.add(layers.Dropout(0.5))
    model.add(layers.Flatten())
    model.add(layers.Dense(64,activation='relu'))
    model.add(layers.Dropout(0.5))
return model
#-----------------------------------
def get_model():
    conv1D = get_1dconv()
    conv2D = get_2dconv()
    t_series_input = layers.Input(shape=(186,6))
    image_input = layers.Input(shape=(80,186,1))

    inp1 = conv1D(t_series_input)
    inp2 = conv2D(image_input)

    merged = layers.Concatenate()([inp1,inp2])

    out = layers.Dense(5, activation='softmax')(merged)

    model = models.Model(inputs=[t_series_input,image_input], outputs=out)
    model.compile(optimizer =
        'RMSprop',loss='categorical_crossentropy',metrics=['accuracy'])
return model
```

The following performance is obtained:

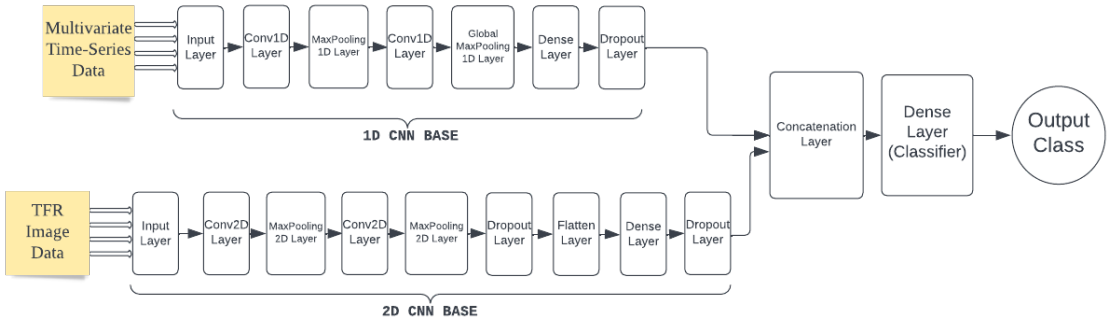


Fig. Two-Input Classifier Model

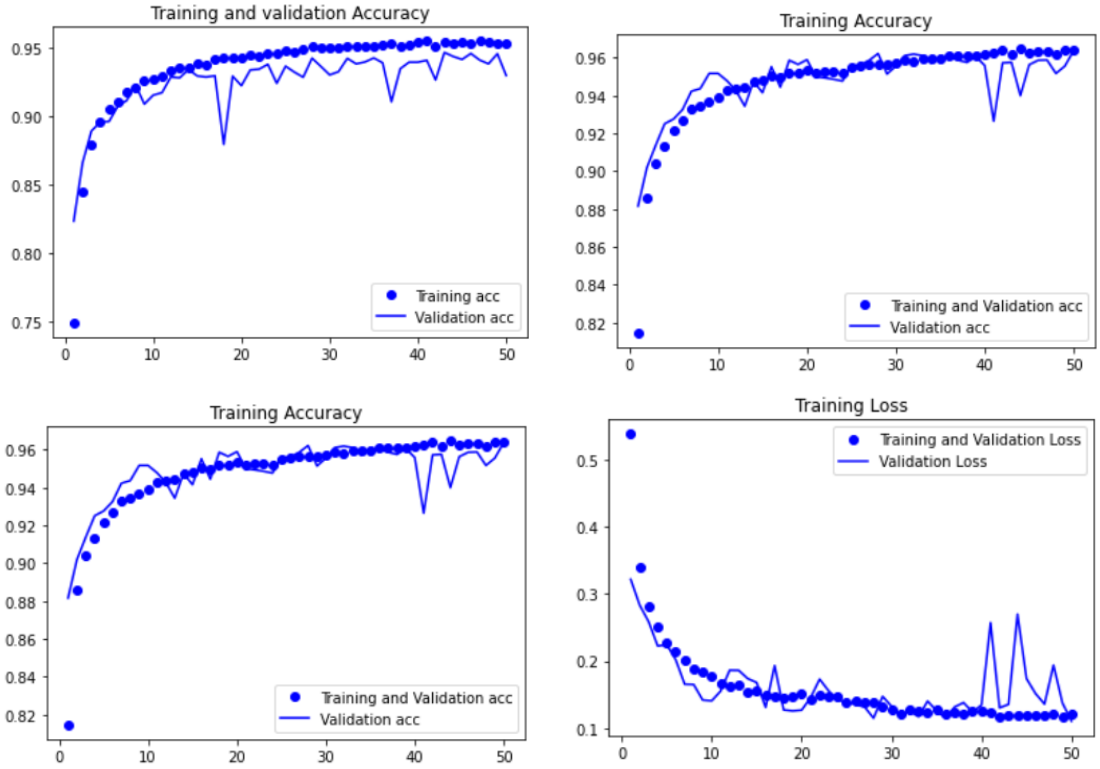


Figure 20: EMD vs FDM, performance using Dual Input Neural Network with Data Augmentation

- **With EMD:** The 4-fold cross validation scores obtained were (in the form of [loss, accuracy]) [[0.2016, 0.9440], [0.1600, 0.9544], [0.1891, 0.9537], [0.2060, 0.9464]] with the average of [0.1892, 0.9496]. Hence, the average validation accuracy of 94.96% is obtained.
  On test data, the performance obtained were [0.2963, 0.9314].

- **With FDM:** The 4-fold cross validation scores obtained were (in the form of [loss, accuracy]) [[0.1336, 0.9603], [0.1642, 0.9635], [0.1311, 0.9602], [0.1764, 0.9546]] with

the average of [0.1513, 0.9596]. Hence, the average validation accuracy of 95.96% is obtained.
On test data, the performance obtained were [0.1771, 0.9603].

- **Comparison:** Again, the performance is competitive, with FDM providing slightly better performance than EMD. Also, a marginal improvement in performance is obtained when compared with 1D CNN classification.

5. Conclusion

In our project, we saw that FDM had provied us with a better performance in Heart Disease Classification than EMD.

The result obtained using 2D CNN can be correlated with the comparison of TFRs using the two methods where we saw that TFR of EMD was very light while that of FDM was darker and more definitive. Hence, in Image classification using 2D CNN, FDM gave far better results.

As far as Multivariate Time Series Classification is concerned, Both EMD and FDM provides us with set of decomposed intrinsic functions. There's no clear cut clarity in which among the two set of decomposed intrinsic functions is a better representation. This can also be figured with the competative results obtained using 1D CNN and Dual Input Neural Network

The Basis of choosing between Empirical Mode Decomposition and Fourier Decomposition Method depends,in general, on two factors. Firstly, on the type of application. In our case, the application was on the ECG signals where we were able to obtain better results with FDM. In some other application, EMD could highly outperform FDM. Secondly, the underlying use of model. We saw that performance of FDM with 2D CNN was far superior than that of EMD. While,performance with 1D CNN and Dual Input Neural Network is highly competative.

6. Refrences

References

[1] Naveed Rehman and Danilo P Mandic. "Multivariate empirical mode decomposition". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 466.2117 (2010), pp. 1291–1302.

[2] Pushpendra Singh et al. "The Fourier decomposition method for nonlinear and non-stationary time series analysis". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2199 (2017), p. 20160871.

[3] Han Li et al. "Dual-input neural network integrating feature extraction and deep learning for coronary artery disease detection using electrocardiogram and phonocardiogram". In: *IEEE Access* 7 (2019), pp. 146457–146469.

[4] Kaggle. *MIT-BIH Dataset*. URL: `https://www.kaggle.com/datasets/shayanfazeli/heartbeat`.

[5] keras. *Dual Input Time Series Classification*. URL: `https://keras.io/examples/timeseries/timeseries_classification_from_scratch/`.

[6] Mathworks. *4-folds cross validation*. URL: `https://www.mathworks.com/help/fuzzy/fis-parameter-optimization-with-k-fold-cross-validation.html`.

[7] Andrew Quinn. *EMD: Empirical Mode Decomposition*. URL: `https://emd.readthedocs.io/en/stable/`.

[8] Wikipedia. *Electrocardiography*. URL: `https://en.wikipedia.org/wiki/Electrocardiography`.

[9] Wikipedia. *Stationary process*. URL: `https://en.wikipedia.org/wiki/Stationary_process`.