Diagnosis of Neuro-degenerative Diseases Using Probabilistic Neural Network

A thesis submitted in partial fulfilment of the
Requirements for the degree of

Bachelor of Technology
In
Bio-Medical Engineering
By

Varsha Agarwalla (111BM0001)
Saswat Kumar Padhy (111BM0528)

Department of Biotechnology and Medical Engineering
National Institute of Technology, Rourkela
Rourkela, Odisha – 769008

NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA 769008, INDIA

Certificate of Approval

This is to certify that the thesis entitled "**Diagnosis of Neuro-degenerative Diseases Using Probabilistic Neural Network**" submitted to the National Institute of Technology, Rourkela by **VARSHA AGARWALLA**, **Roll No. 111BM0001** and **SASWAT PADHY**, **Roll No. 111BM0528** is a record of an original work carried out under my supervision and guidance. The outcome presented in this thesis has not been, to the best of my insight, submitted to some other University or Institute for the honour of any degree or recognition. In my opinion, the thesis has reached the standards fulfilling the requirement for the award of the degree of Bachelor of Technology in accordance with regulations of the Institute.

Prof. K Pramanik
Professor
Department of Biotechnology & Medical Engineering
National Institute of Technology, Rourkela

Date: 11-05-2015

Acknowledgement

This thesis has been possible due to the help and endeavour of many people.

Foremost, we would like to express our appreciation and thanks to our project advisor, Prof. K Pramanik, whose mentor-ship has been paramount, not only in carrying out the research for this thesis, but also in developing long-term goals for our career. Her guidance has been unique and delightful. She provided her able guidance whenever we needed it. Yet she always motivated us to be an independent thinker, and to choose and work with independence.

We would also like to express our gratitude towards our project review panel for their attention. The constructive feedback received has been keenly instrumental in improvising our work further.

We would also like to thank our companions for their encouragement and whole-hearted cooperation.

Varsha Agarwalla

Saswat Kumar Padhy

Abstract

In recent years researchers have been given attention to present a non-invasive approach to deal with the diseases quickly and all the more unequivocally. The major cause of death of patients is due to the wrong diagnosis. Thus a right method is to be identified which would help the physicians to concentrate on the cause of illness and its diagnosis so as to abstain from squandering valuable time- that may be precious for the patient- on diagnosis. In this particular thesis, we have tried to build up a computerized methodology to deal with patients having problems in walking by analysing their gait signal. We chose four groups of patients, namely patients suffering from neuro-degenerative diseases such as Huntington's disease, Parkinson's disease and Amyotrophic Lateral Sclerosis and a group of healthy control subjects. So we have utilized Probabilistic Neural Network (PNN) as a classifier, to identify or differentiate the different patients and define the type of disease that they are suffering from, with accuracy in the range of 85- 95% so as to diagnose the diseases correctly and as a result, minimize the death rates.

Keywords-Probabilistic Neural Network, Neuro-Degenerative Diseases, Gait Signals

Contents

List of Tables and Figures

Chapter 1

Introduction

It is quite essential that the different diseases are identified and examined quickly so that correct treatment could be provided for the same. Some patients are known to be dead due to lack of proper examination [2] Thus, we look for a non-invasive approach. Moreover, we can reduce the amount and time taken for correct diagnosis if we could identify the disease in a simple way. To identify the cause of disorder related to movement, we can use the gait signal.

This thesis presents an efficient approach for classification based on the features that has been extracted from the gait signal of patients suffering with three different locomotive disorders, namely Huntington's disease, Parkinson's and Amyotrophic Lateral Sclerosis(ALS) along with healthy control subjects. These locomotive disorders are due to the wrong functioning of the brain and hence, are also known as neuro-degenerative diseases. All these three locomotive diseases are nevertheless same in nature but with slight differences.

Patients with Huntington's disease face the problem of co-ordination in their body movements. This particular disease is hereditary and it reduces the life expectancy. Patients with Parkinson's disease have their motor skills affected and thus can be identified with muscles that have become rigid, body movement slows down and in quite extreme case the person might lose his complete physical movement. Very old people having age near to 85 years have at least one symptom of Parkinsonism [1]. Amyotrophic Lateral Sclerosis is a type of disorder in which both the upper and lower motor neurons of the patients degenerate and the muscles become weak and hence fail to send signals, losing the ability to control the movement of the muscles voluntarily, leading to atrophy. It is a chronic disease. By the year 2040, these three diseases are identified to beat cancer and would be the second most reason of death among the elderly [3].

1.1 Signal and its Types

In image processing, a signal is a physical quantity which varies with space and contains information about space. Signals can extensively be classified into stationary and non-stationary. An image is a non-stationary signal. To examine a non-stationary signal, multi-resolution techniques are required. Many multi-resolution techniques exist. Some of them are:

1. Short Time Fourier Transform

2. Wavelet Transform

Wavelet Transform has many advantages over Short Time Fourier Transform.

1.2 Wavelet Transform

The Continuous Wavelet Transform can be defined as a series of correlations of the time series with a function called a wavelet:

$$W(\tau, d) = \int_{-\infty}^{\infty} h(t) w(t - \tau, d) dt$$

A wavelet is defined as an oscillatory wave with its amplitude starting at zero, increases above and then decreases back to zero. The complicated data like sound signal and images are decomposed and remade with high exactness [4]. To extract features from various signals, Wavelet transform is used widely. A discrete wavelet transform is a multi-resolution decomposition method in which a particular image is decomposed into detail and approximate images. These detail and approximate images are matrices with wavelet co-efficient values and each co-efficient is regarded as the original image's feature. Wavelet Transform is the improved version of Fourier Transform.

1. Fourier transform uses Fourier series for representing functions as a sum of sinusoidal functions; whereas to describe a non-continuous function, we prefer wavelets

2. Fourier basis functions are located in the frequency domain but not in the time domain; on the other hand, bases of wavelet functions are well located in the frequency (via dilations) and the time domain (via translations). Any small changes in the frequency component in the Fourier transform will give rise to changes everywhere in the time domain

1.3 Approximate Entropy

To analyse the time-series data researchers use approximate entropy. It basically measures the system's complexity. To calculate the entropy accurately, data will be required in large amount and the result obtained is affected by the noise level of the system. Therefore, for experimental data, entropy calculation method is not used and hence ApEn is created. For a time-series data, ApEn is used for regularity and unconventionality changes. ApEn mirrors the probability that ``similar" observation patterns will not be trailed by extra "similar" observations [8]. The algorithm [9] that is used to calculate ApEn of a sequence S_N having N instantaneous heart-rate estimations, two parameters m and r are chosen. ApEn is given as ApEn(s_N, m, r). Pattern length is given by m and r denotes the similarity criteria.

For a pattern of length m, that has begun at i within the sequence, is indicated by $P_m(i)$. Similarly, for the one that has begun at j, can be indicated as $P_m(j)$. For any pair of relating estimations in the pattern, these two are said to be comparable if only the value is less than r. $C_m(r)$ can be calculated as the ratio of the number of similar patterns of length m by the total number of such patterns of length m that exists in the whole sequence (N-m+1).

$$C_m(r) = \frac{n_m(r)}{N-m+1}$$

Thus the ApEn of the entire sequence S_N for pattern length m and similarity criteria r is computed as:

$$\text{ApEn}\ (s_N, m, r) = \ln\left[\frac{C_m(r)}{C_{m+1}(r)}\right]$$

If the ApEn value is small, it can be said that the observation values are very similar. Whereas for a highly unpredictable series, for example a time-series database, such similar patterns would not be obtained and hence the ApEn value will be quite large.

1.4 Probability Neural Network

Neural networks with their exceptional capacity to determine significance from complicated data can be utilized to recognize patterns that are so intricate it would be impossible to be recognized by either people or other computer strategies [6]. Probability Neural Network (PNN) is a feed-forward Artificial Neural Network in which an input pattern is forwarded from one layer to the next without any feed back to the previous layers. An ordinary PNN has four layers, namely input, pattern, summation and output. The input units supply the same values to all pattern units. The pattern units form a dot product between the input pattern vector x and a weighted vector w_i ($z_i = x w_i$), dividing the result by square of standard deviation, which is followed by the nonlinear neuron activation function:

$$g(z_i) = \exp\left(-\frac{(w_i - x)^1 (w_i - x)}{2\sigma^2}\right)$$

This Bayesian function considers the relative probability of events and uses from the earlier data to enhance the prediction. The summation units basically aggregate the inputs from the pattern units, relating to the classification from which the training patterns were chosen. Rehashing this technique for every class, the un-normalized density functions gk (x), for k=1, 2… K were evaluated. The Bayesian likelihood that the case was from class k is as per the following:

$$P(x \epsilon\ k) = \frac{g_i(x)}{\sum_{i=1}^{k} g_i\,(x)}$$

The output unit picks up the maximum of the probabilities and produces the class that each belongs to.

A classifier is constructed by integrating the Discrete Fourier Transform with the PNN model. At first, the DWT is used to exact the necessary features from the time series data. Then PNN classifies these features to identify the type of disease that the person is suffering from.

Chapter 2

Literature Review

Mantzaris, D.H. *et al.* proposed the use of the first computational technique, utilizing ANN for the study of osteoporosis. Both MLP and PNN were used. Different architectures of MLP were used by changing the number of nodes in the hidden layer. Similarly, PNN was also used by changing the spread coefficient value ranging from 0.1 to 50 and obtained greater accuracy with PNN [10]. The PNN was considered as the suitable method in comparison to MLP for predicting the risk associated with osteoporosis and this computation technique was proved to be far better than the tests which are carried out in labs to measure the bone density [10].

Mohammad Mikaili *et al.* attempted to build a methodology for diagnosing patients with locomotion problems by reading their gait signal features. They chose four groups, namely Huntington's disease patients, and patients suffering with Parkinson disease and Amyotrophic Lateral Sclerosis patients and also considered few healthy people. 17 well-known classifiers were used and accuracy was calculated for each classifier and obtained accuracy around 86.95% with the use of Quadratic Bayes Normal Classifier and they obtained the least of accuracy with the Logistic and nearest mean type of classifiers [11].

Sheldon R. Simon *et al.* proposed the application of analytical techniques to test a patient's leg movements and would enormously upgrade the capability of gait labs and as a result minimise the time.
Due to lack of budgets in hospitals and difficulty in analyzing the gait signal, gait labs are considered as unproductive. So they suggested the idea of utilizing sensors which can be fitted on the legs to measure the ground reaction force when a patient keeps his foot during walking rather than using bio-markers for the same [12].

Weijun Tao *et al.* proposed the use of wearable sensors to analyse the gait signal. They used these sensors for long time and with ease and great mobility and suggested its use along with applying an analytical algorithm so as to design a system for future gait analysis [13].

Resul Das used four different classification methods, namely Neural Networks, DM neural, Regression and Decision Tree for diagnosing Parkinson's disease and calculated the score of

each classifier. He obtained the best score of around 92% with the application of Neural Networks[14] .

Wei-Hsin Wang *et al.* had built an algorithm to detect the stride of the patients and obtained gait information which they used to select the most significant feature. Later, they used Probabilistic Neural Network (PNN) to classify if the patients suffer from Alzheimer's disease or not and obtained an accuracy of around 63.33% in women group and 70% in men group [15].
Shouman, M. *et al* proposed the use of analytical tools to extract useful knowledge from the huge amount of medical database. Since, heart disease has been the reason of death in the past 10 years so they used a strong data mining technique so as to help the doctors in the correct treatment of the disease which was previously not paid heed to [16].

So their approach has given an idea to design a particular classifier type which would help to diagnose the patients having difficulty in walking, fast and more accurately. So an attempt has been made to introduce a method of identification of the class that the patient belongs to, thereby, increasing the accuracy and hence developing a better computational technique which could be used in preventing wrong diagnosis and for the proper treatment of the people minimising their death rates.

Chapter 3

Methodology

The following step-by-step methodology was followed for the proper classification of neuro-degenerative diseases.

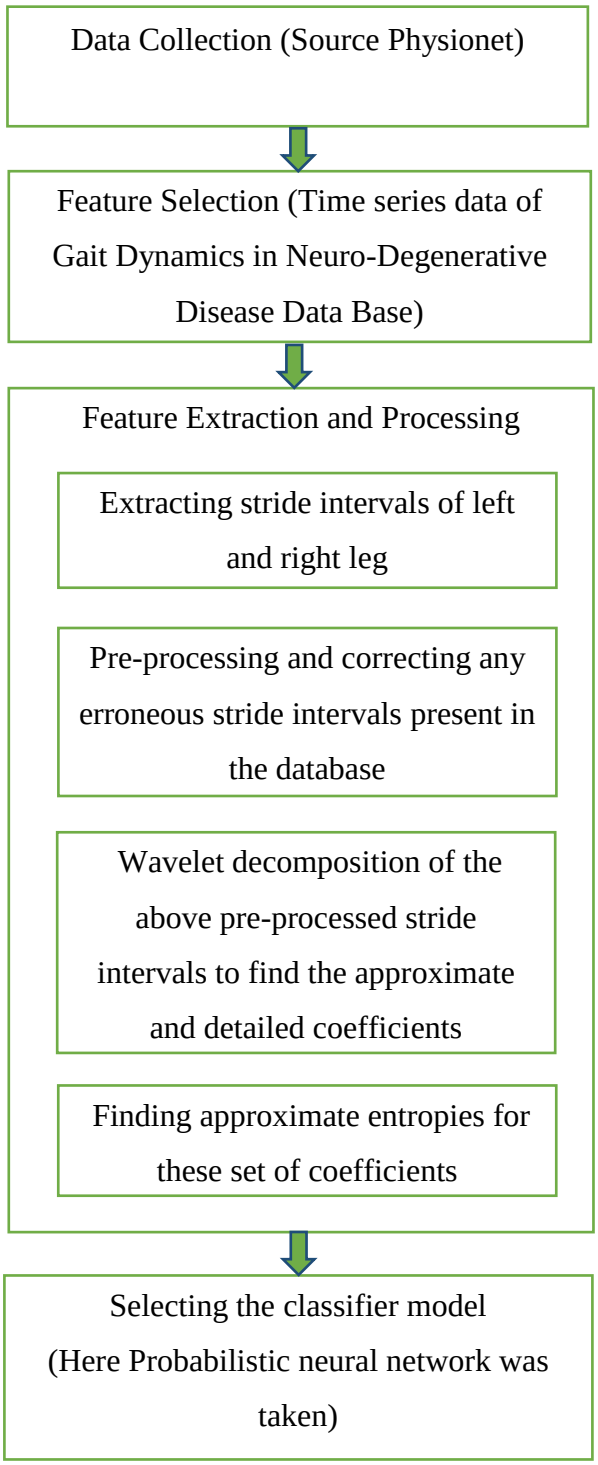

Fig.1 Block diagram of proposed methodology

3.1 Data Collection

For studying the gait dynamics of the subjects with neuro-degenerative diseases, gait data base is taken from the PhysioNet database of MIT

3.2 Feature Selection

- Time series data of Gait Dynamics in Neuro-Degenerative Disease Database(gaitndd) (.ts files) is downloaded
- The set contained
    - ✓ 13 Amyotrophic lateral sclerosis patients' gait-data
    - ✓ 15 Parkinson's disease patients' gait-data
    - ✓ 20 Huntington's disease patients' gait-data
    - ✓ 16 healthy control subjects' gait-data
- For example

als6.ts

22.7167	1.6400	1.5067	0.4667	0.4100	28.46	27.21	1.1733	1.0967	71.54	72.79	0.7633	46.54
24.3367	1.6200	1.7167	0.5033	0.4767	31.07	27.77	1.1167	1.2400	68.93	72.23	0.6400	39.51
25.9567	1.6200	1.6033	0.4933	0.4733	30.45	29.52	1.1267	1.1300	69.55	70.48	0.6533	40.33
27.6500	1.6933	1.5733	0.5267	0.3967	31.10	25.21	1.1667	1.1767	68.90	74.79	0.7700	45.47
29.2433	1.5933	1.7267	0.4767	0.4033	29.92	23.36	1.1167	1.3233	70.08	76.64	0.7133	44.77
30.8467	1.6033	1.5467	0.5167	0.3767	32.22	24.35	1.0867	1.1700	67.78	75.65	0.7100	44.28
32.4767	1.6300	1.6300	0.4567	0.4133	28.02	25.36	1.1733	1.2167	71.98	74.64	0.7600	46.63
34.0900	1.6133	1.6633	0.4400	0.3900	27.27	23.45	1.1733	1.2733	72.73	76.55	0.7833	48.55
35.7067	1.6167	1.6167	0.5133	0.4667	31.75	28.87	1.1033	1.1500	68.25	71.13	0.6367	39.38
37.4467	1.7400	1.6767	0.4900	0.4967	28.16	29.62	1.2500	1.1800	71.84	70.38	0.7533	43.30
39.0133	1.5667	1.6700	0.4733	0.4633	30.21	27.74	1.0933	1.2067	69.79	72.26	0.6300	40.21
40.5733	1.5600	1.4833	0.4867	0.3900	31.20	26.29	1.0733	1.0933	68.80	73.71	0.6833	43.80
42.1900	1.6167	1.6067	0.5200	0.4067	32.16	25.31	1.0967	1.2000	67.84	74.69	0.6900	42.68
43.6267	1.4367	1.6033	0.4200	0.4467	29.23	27.86	1.0167	1.1567	70.77	72.14	0.5700	39.68
45.2500	1.6233	1.4800	0.4767	0.4933	29.36	33.33	1.1467	0.9867	70.64	66.67	0.6533	40.25
46.7767	1.5267	1.5500	0.5100	0.4000	33.41	25.81	1.0167	1.1500	66.59	74.19	0.6167	40.39
48.3133	1.5367	1.5467	0.4500	0.4300	29.28	27.80	1.0867	1.1167	70.72	72.20	0.6567	42.73
49.9433	1.6300	1.6067	0.4433	0.4400	27.20	27.39	1.1867	1.1667	72.80	72.61	0.7467	45.81
51.5067	1.5633	1.5467	0.5067	0.4333	32.41	28.02	1.0567	1.1133	67.59	71.98	0.6233	39.87
52.9933	1.4867	1.5233	0.4500	0.4067	30.27	26.70	1.0367	1.1167	69.73	73.30	0.6300	42.38
54.5200	1.5267	1.5000	0.4700	0.3933	30.79	26.22	1.0567	1.1067	69.21	73.78	0.6633	43.45
56.0300	1.5100	1.5467	0.4767	0.3967	31.57	25.65	1.0333	1.1500	68.43	74.35	0.6367	42.16
57.5400	1.5100	1.5167	0.4200	0.4033	27.81	26.59	1.0900	1.1133	72.19	73.41	0.6867	45.47
59.1133	1.5733	1.5300	0.4933	0.4467	31.36	29.19	1.0800	1.0833	68.64	70.81	0.6333	40.25

Fig.2 Time-series data of an ALS patient

In these cases, als6.ts as in fig.1, there are 13 columns each representing some features such as:-

1	Elapsed Time (sec)
2	Left Stride Interval (sec)
3	Right Stride Interval (sec)
4	Left Swing Interval (sec)
5	Right Swing Interval (sec)
6	Left Swing Interval (% of stride)
7	Right Swing Interval (% of stride)
8	Left Stance Interval (sec)
9	Right Stance Interval (sec)
10	Left Stance Interval (% of stride)
11	Right Stance Interval (% of stride)
12	Double Support Interval (sec)
13	Double Support Interval (% of stride)

1. **Representation of all features of a time-series data**

3.3 Feature Extraction and Processing

The left stride intervals and right stride intervals of each time-series file are extracted and taken into account for further processing

3.3.1 Pre-processing and correcting any erroneous stride intervals present in the database

Some erroneous stride intervals present in time series files are corrected. The MATLAB code designed for it:-

```
%clc
%clear all
%load park9.ts;
%x1=park9(:,2);
%x2=park9(:,3);
%disp('park9')
    s1=0;
    s2=0;
if x1(1)>2
x1(1)=1.5;
end
if x2(1)>2
x2(1)=1.5;
end
for k1=1:length(x1)
if x1(k1)>2
x1(k1)=x1(k1-1);
end
if x2(k1)>2
x2(k1)=x2(k1-1);
end
end
for j=1:length(x1)
        s1=s1+x1(j);
        s2=s2+x2(j);
end
    avgs1=s1/length(x1);
    avgs2=s2/length(x2);
if x1(1)>(avgs1+0.3)
x1(1)=1.5;
end
if x2(1)>(avgs2+0.3)
x2(1)=1.5;
end
fori=1:length(x2)
if avgs1<1.4
if x1(i)>1.5
x1(i)=x1(i-1);
end
else
if x1(i)>(avgs1+0.3)
x1(i)=x1(i-1);
end
end
if avgs2<1.4
if x2(i)>1.5
x2(i)=x2(i-1);
end
else
if x1(i)>(avgs2+0.3)
x1(i)=x1(i-1);
end
end
end
```

(forloop.m)

x1 and x2 are 2nd and 3rd column of a time series file (in this case park9.ts) which represent stride intervals of left and right leg respectively. The stride intervals should be nearly uniform after any elapsed time intervals and should be within 1s- 1.5s in general for all cases. Otherwise there will be an undesirable sharp increase in peak for stride intervals vs. elapsed time plot. So this program brought down any undesirable stride interval down to its previous stride value on basis of the average stride interval or those with above 2s stride values down to 1.5s(assumption). Load commands are given under comment sections for better understanding. This command has been used in later code and forloop.m is called.

3.3.2 Wavelet decomposition of these pre-processed stride intervals to find the approximate and detailed coefficients

An m-file is created named "wavelet.m", in which wavelet decomposition of both left and right stride intervals has been done at level 5 and 6 for each patients. Hence approximate coefficients and detail coefficients are found for each patient. The MATLAB code "wavelet.m" is:-

```
forloop

subplot(1,2,1), plot(x1), title('Left
Stride Interval Series')


subplot(1,2,2), plot(x2),
title('Right Stride Interval Series')
[C,L] = wavedec(x1,6,'db6');

 A5 =appcoef(C,L,'db6',5);
 A6 =appcoef(C,L,'db6',6);

 D5 =detcoef(C,L,5);
 D6 =detcoef(C,L,6);

 figure
 subplot(2,2,1)
```

```
plot(A5),title('Pleft-appcoef5')
subplot(2,2,2)
plot(A6),title('Pleft-appcoef6')
subplot(2,2,3)
plot(D5),title('Pleft-detcoef5')
subplot(2,2,4)
plot(D6),title('Pleft-detcoef6')

n (1)=fix(length(A5)/10);
n (2)=fix(length(A6)/10);
n (3)=fix(length(D5)/10);
n (4)=fix(length(D6)/10);
n= [n(1),n(2),n(3),n(4)];
fori=1:4
if n(i)<2
n(i)=2;
end
end
r1=(max(A5)-min(A5))/n(1);
apna1l = apen(n(1),r1,A5);

r2=(max(A6)-min(A6))/n(2);
apna2l = apen(n(2),r2,A6);

r3=(max(D5)-min(D5))/n(3);
apnd1l = apen(n(3),r3,D5);

r4=(max(D6)-min(D6))/n(4);
apnd2l = apen(n(4),r4,D6);
PLeft=[apna1l apna2l apnd1l apnd2l]

[C,L] = wavedec(x2,6,'db6');

 A5 =appcoef(C,L,'db6',5);
 A6 =appcoef(C,L,'db6',6);

 D5 =detcoef(C,L,5);
 D6 =detcoef(C,L,6);

figure
subplot(2,2,1)
plot(A5),title('PRight-appcoef5')
subplot(2,2,2)
plot(A6),title('PRight-appcoef6')
subplot(2,2,3)
plot(D5),title('PRight-detcoef5')
subplot(2,2,4)
plot(D6),title('PRight-detcoef6')

n(1)=fix(length(A5)/10);
n(2)=fix(length(A6)/10);
n(3)=fix(length(D6)/10);
n(4)=fix(length(D6)/10);
n=[n(1),n(2),n(3),n(4)];
```

```
fori=1:4
if n(i)<2
n(i)=2;
end
end

r1=(max(A5)-min(A5))/n(1);
apna1r = apen(n(1),r1,A5);

r2=(max(A6)-min(A6))/n(2);
apna2r = apen(n(2),r2,A6);

r3=(max(D5)-min(D5))/n(3);
apnd1r = apen(n(3),r3,D5);

r4=(max(D6)-min(D6))/n(4);
apnd2r = apen(n(4),r4,D6);

PRight = [apna1r apna2r apnd1r apnd2r]
```

(wave.m)

wavedec is inbuilt MATLAB function that performs wavelet decomposition of any signal(here x1 and x2) up to any level(here level 6) using a specified wavelet(here db6).The outputs of decomposition would be D6, D5, D4, D3,D2,D1,A6,L . The process of decomposition up to level 3 is as shown in Fig.2

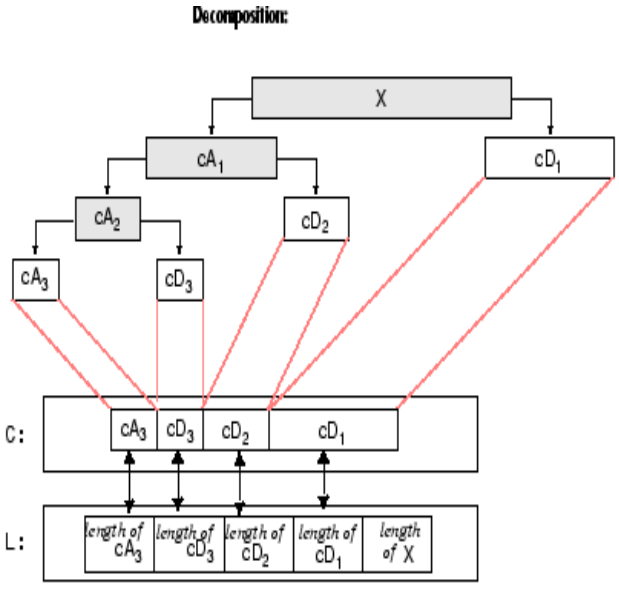

Fig.3 The wavelet decomposition up to level 3

It is found that the detailed coefficients are not further decomposed whereas the approximate coefficients are decomposed again to detailed and approximate coefficients.

- Each D6, A6, etc. has a series of coefficients value. So C is a two dimensional array having coefficients value and L is a vector consisting of corresponding length of coefficient value number

- The required coefficient value number A6,A5,D5,D6 for both left and right stride intervals are extracted using appcoef and detcoef respectively

- For approximate entropic calculations.
  - ✓ A5, A6, D5, D6 form data sets
  - ✓ n1, n2, n3, n4 are defined as length of A5, A6, D5, D6 each divided by 10. These form the window length. As the window length should be greater than or equal to 2 so any n values less than 2 is brought up to 2.
  - ✓ r1, r2, r3 and r4 are their corresponding difference criteria and is defined as difference between maximum coefficient value to minimum coefficient value of a given coefficient number divided by its corresponding n values.

3.3.3 Finding approximate entropies for these set of coefficients

```
functionapen = apen(n,r,a)
data =a;

for m=n:n+1; % run it twice, with window size differing by 1

set = 0;
count = 0;
counter = 0;

fori=1:(length(data))-m+1,
current_window = data(i:i+m-1); % current window stores the sequence to
be compared with other sequences

for j=1:length(data)-m+1,
sliding_window = data(j:j+m-1); % get a window for comparison with the
current_window

for k=1:m,
if((abs(current_window(k)-sliding_window(k))>r) && set == 0)
set = 1; % i.e. the difference between the two sequence is greater than
the given value
end
end

if(set==0)
count = count+1; % this measures how many sliding_windows are similar to
the current_window
end

set = 0; % resetting 'set'

end

counter(i)=count/(length(data)-m+1); % we need the number of similar
windows for every cuurent_window
count=0;
i;
end%  for i=1:(length(data))-m+1, ends here

counter;  % this tells how many similar windows are present for each
window of length m
%total_similar_windows = sum(counter);
%window_correlation = counter/(length(data)-m+1);

correlation(m-n+1) = ((sum(counter))/(length(data)-m+1));
end% for m=n:n+1; % run it twice
correlation(1);
correlation(2);
apen = log(correlation(1)/correlation(2));
```

- The 3 parameters of each coefficient number N,R and A as in wave.m are fed to apen.m function which is called to find the approximate entropies of these coefficients.

- Hence PLeft (contains 4 entropy values for left leg) and PRight (contains 4 entropy values for right leg) were found for each patients in the following serial.
    - ✓ Entropy value approximate coefficient of level-5 using db6-decomposition of wavelet for left stride interval
    - ✓ Entropy value approximate coefficient of level-6 using db6-decomposition of wavelet for left stride interval
    - ✓ Entropy value detailed coefficient of level-5 using db6-decomposition of wavelet for left stride interval
    - ✓ Entropy value detailed coefficient of level-6 using db6-decomposition of wavelet for left stride interval
    - ✓ Entropy value approximate coefficient of level-5 using db6-decomposition of wavelet for right stride interval
    - ✓ Entropy value approximate coefficient of level-6 using db6-decomposition of wavelet for right stride interval
    - ✓ Entropy value detailed coefficient of level-5 using db6-decomposition of wavelet for right stride interval
    - ✓ Entropy value detailed coefficient of level-6 using db6-decomposition of wavelet for right stride interval

- Output entropic values obtained for each time series files were written in a new data sheet

For ALS

0.1834	0.0183	0.2532	0.2795	0.1834	0.2865	0.2022	0.2917;
0.2299	0.2213	0.2141	0.2454	0.2377	0.2213	0.2118	0.2632;
0.0616	0.0631	0.2794	0.1248	0.0374	0.0315	0.1997	0.0631;
0.0126	0.0178	0.2537	0.2770	0.0126	0.0178	0.2049	0.2770;
0.0998	0.0579	0.2366	0.2370	0.1307	0.1171	0.1517	0.1408;
0.3230	0.1227	0.2543	0.2482	0.3300	0.0195	0.2599	0.2254;

0.1869	0.2363	0.2705	0.1137	0.1897	0.2865	0.0232	0.2472;
0.0766	0.0631	0.1945	0.2201	0.0629	0.0631	0.2006	0.2068;
0.1258	0.0631	0.0123	0.0469	0.1258	0.0631	0.0096	0.0671;
0.3299	0.2632	0.1712	0.2123	0.3874	0.1909	0.1524	0.2123;
0.1886	0.1909	0.1318	0.3168	0.1130	0.1909	0.1215	0.3407;
0.1023	0.0922	0.1779	0.2148	0.0863	0.0922	0.1779	0.2148;
0.0935	0.0756	0.1740	0.2463	0.0812	0.0756	0.2595	0.1643;

For Huntington

0.2154	0.0816	0.1490	0.1814	0.0650	0.0169	0.1294	0.2318;
0.0560	0.0579	0.1520	0.2221	0.0774	0.0579	0.1299	0.3032;
0.1278	0.0839	0.1735	0.2201	0.1037	0.0631	0.1361	0.2642;
0.1982	0.0636	0.1352	0.2294	0.1236	0.0232	0.2328	0.1957;
0.0729	0.0631	0.2299	0.2454	0.0624	0.0539	0.2668	0.0958;
0.0289	0.0126	0.2377	0.2201	0.0289	0.0126	0.1575	0.2290;
0.1086	0.0579	0.1827	0.2087	0.1002	0.2546	0.1130	0.2551;
0.2207	0.1004	0.1495	0.2213	0.2207	0.0631	0.1131	0.1909;
0.1244	0.1469	0.1963	0.0929	0.1244	0.1469	0.1734	0.1058;
0.0456	0.0872	0.1797	0.2454	0.0506	0.0819	0.1874	0.2028;
0.1361	0.0878	0.1558	0.1303	0.1581	0.0363	0.2184	0.1039;
0.2648	0.1584	0.1618	0.1303	0.1442	0.0594	0.1709	0.1748;
0.0660	0.0321	0.1957	0.2159	0.0108	0.0321	0.2043	0.2363;
0.0276	0.0126	0.0948	0.2213	0.0289	0.0126	0.1618	0.1954;
0.0175	0.0207	0.1934	0.1647	0.0175	0.0207	0.1666	0.2366;
0.0089	0.0149	0.1879	0.3128	0.0404	0.0195	0.1727	0.2880;
0.1402	0.2551	0.0181	0.1606	0.1335	0.1354	0.0389	0.1746;
0.1179	0.0126	0.1358	0.1139	0.0316	0.0126	0.2736	0.2049;
0.1709	0.0631	0.2842	0.4213	0.1680	0.0382	0.1347	0.2762;

For Control

0.1467	0.2272	0.1172	0.2454	0.1972	0.0863	0.1219	0.2272;
0.1088	0.0126	0.2022	0.1978	0.2898	0.1623	0.2362	0.1814;
0.1501	0.1464	0.1219	0.1954	0.1474	0.2272	0.1107	0.1927;
0.1567	0.0854	0.1477	0.1587	0.2021	0.1257	0.1477	0.1639;
0.0070	0.0180	0.2252	0.1684	0.0105	0.0126	0.2560	0.1978;
0.0257	0.0232	0.1529	0.2125	0.0257	0.0232	0.1440	0.1814;
0.1866	0.1321	0.1813	0.2642	0.1380	0.1400	0.2022	0.2172;
0.1402	0.0126	0.1841	0.1584	0.0687	0.0126	0.1915	0.1951;
0.1319	0.0341	0.1414	0.2007	0.1202	0.0396	0.1339	0.2007;
0.0621	0.0232	0.1040	0.1772	0.1658	0.0232	0.0876	0.1772;
0.2742	0.1469	0.1398	0.1807	0.2808	0.1469	0.1445	0.1807;

0.0452	0.0176	0.0673	0.1909	0.0649	0.0176	0.0760	0.1909;
0.1808	0.1814	0.1802	0.2213	0.1562	0.2106	0.1680	0.2213;
0.3066	0.1584	0.2594	0.2551	0.0208	0.0126	0.2279	0.2281;
0.1024	0.1227	0.0717	0.2568	0.1071	0.0583	0.1058	0.2568;
0.1384	0.0238	0.1616	0.2106	0.1411	0.1249	0.1585	0.1853;

For Parkinson

0.1711	0.3254	0.2008	0.1408	0.1358	0.0447	0.3176	0.0559;
0.1473	0.1395	0.1315	0.1639	0.1332	0.1116	0.1557	0.1587;
0.2803	0.3254	0.0794	0.0757	0.3128	0.3203	0.1199	0.2454;
0.3409	0.2090	0.1980	0.2172	0.1526	0.1840	0.2623	0.2201;
0.1614	0.0176	0.1558	0.2290	0.0410	0.0063	0.1148	0.1354;
0.1172	0.1193	0.1907	0.0490	0.1304	0.1193	0.2345	0.0671;
0.0187	0.0277	0.1692	0.2213	0.0395	0.0277	0.1816	0.2213;
0.0654	0.0238	0.2131	0.0900	0.1688	0.0872	0.2276	0.0341;
0.1412	0.1400	0.2086	0.1439	0.1586	0.1400	0.1451	0.1372;
0.0172	0.1529	0.2048	0.1693	0.1364	0.2007	0.1297	0.1483;
0.2017	0.0819	0.0689	0.2106	0.1656	0.2213	0.1050	0.2123;
0.3176	0.1414	0.1802	0.2172	0.2280	0.1050	0.2182	0.0972;
0.0173	0.0126	0.2288	0.1882	0.0439	0.0126	0.1562	0.2454;
0.0943	0.0726	0.1297	0.2020	0.0885	0.0726	0.1445	0.1729;
0.0415	0.0208	0.3122	0.2213	0.0222	0.0126	0.2054	0.1909;

3.4 Selecting appropriate classifier (Probabilistic neural network)

- **Case-1: Training set=63 and Testing set=63**

```
clc
clearall
loadpark9.ts;
x1=park9(:,2);
x2=park9(:,3);
disp('park9')
Q=[
```

```
0.1834    0.0183    0.2532    0.2795    0.1834    0.2865    0.2022    0.2917;
0.2299    0.2213    0.2141    0.2454    0.2377    0.2213    0.2118    0.2632;
0.0616    0.0631    0.2794    0.1248    0.0374    0.0315    0.1997    0.0631;
0.0126    0.0178    0.2537    0.2770    0.0126    0.0178    0.2049    0.2770;
0.0998    0.0579    0.2366    0.2370    0.1307    0.1171    0.1517    0.1408;
0.3230    0.1227    0.2543    0.2482    0.3300    0.0195    0.2599    0.2254;
0.1869    0.2363    0.2705    0.1137    0.1897    0.2865    0.0232    0.2472;
0.0766    0.0631    0.1945    0.2201    0.0629    0.0631    0.2006    0.2068;
0.1258    0.0631    0.0123    0.0469    0.1258    0.0631    0.0096    0.0671;
0.3299    0.2632    0.1712    0.2123    0.3874    0.1909    0.1524    0.2123;
0.1886    0.1909    0.1318    0.3168    0.1130    0.1909    0.1215    0.3407;
0.1023    0.0922    0.1779    0.2148    0.0863    0.0922    0.1779    0.2148;
0.0935    0.0756    0.1740    0.2463    0.0812    0.0756    0.2595    0.1643;


0.2154    0.0816    0.1490    0.1814    0.0650    0.0169    0.1294    0.2318;
0.0560    0.0579    0.1520    0.2221    0.0774    0.0579    0.1299    0.3032;
0.1278    0.0839    0.1735    0.2201    0.1037    0.0631    0.1361    0.2642;
0.1982    0.0636    0.1352    0.2294    0.1236    0.0232    0.2328    0.1957;
0.0729    0.0631    0.2299    0.2454    0.0624    0.0539    0.2668    0.0958;
0.0289    0.0126    0.2377    0.2201    0.0289    0.0126    0.1575    0.2290;
0.1086    0.0579    0.1827    0.2087    0.1002    0.2546    0.1130    0.2551;
0.2207    0.1004    0.1495    0.2213    0.2207    0.0631    0.1131    0.1909;
0.1244    0.1469    0.1963    0.0929    0.1244    0.1469    0.1734    0.1058;
0.0456    0.0872    0.1797    0.2454    0.0506    0.0819    0.1874    0.2028;
0.1361    0.0878    0.1558    0.1303    0.1581    0.0363    0.2184    0.1039;
0.2648    0.1584    0.1618    0.1303    0.1442    0.0594    0.1709    0.1748;
0.0660    0.0321    0.1957    0.2159    0.0108    0.0321    0.2043    0.2363;
0.0276    0.0126    0.0948    0.2213    0.0289    0.0126    0.1618    0.1954;
0.0175    0.0207    0.1934    0.1647    0.0175    0.0207    0.1666    0.2366;
0.0089    0.0149    0.1879    0.3128    0.0404    0.0195    0.1727    0.2880;
0.1402    0.2551    0.0181    0.1606    0.1335    0.1354    0.0389    0.1746;
0.1179    0.0126    0.1358    0.1139    0.0316    0.0126    0.2736    0.2049;
0.1709    0.0631    0.2842    0.4213    0.1680    0.0382    0.1347    0.2762;


0.1467    0.2272    0.1172    0.2454    0.1972    0.0863    0.1219    0.2272;
0.1088    0.0126    0.2022    0.1978    0.2898    0.1623    0.2362    0.1814;
0.1501    0.1464    0.1219    0.1954    0.1474    0.2272    0.1107    0.1927;
0.1567    0.0854    0.1477    0.1587    0.2021    0.1257    0.1477    0.1639;
0.0070    0.0180    0.2252    0.1684    0.0105    0.0126    0.2560    0.1978;
0.0257    0.0232    0.1529    0.2125    0.0257    0.0232    0.1440    0.1814;
0.1866    0.1321    0.1813    0.2642    0.1380    0.1400    0.2022    0.2172;
0.1402    0.0126    0.1841    0.1584    0.0687    0.0126    0.1915    0.1951;
0.1319    0.0341    0.1414    0.2007    0.1202    0.0396    0.1339    0.2007;
0.0621    0.0232    0.1040    0.1772    0.1658    0.0232    0.0876    0.1772;
0.2742    0.1469    0.1398    0.1807    0.2808    0.1469    0.1445    0.1807;
0.0452    0.0176    0.0673    0.1909    0.0649    0.0176    0.0760    0.1909;
0.1808    0.1814    0.1802    0.2213    0.1562    0.2106    0.1680    0.2213;
0.3066    0.1584    0.2594    0.2551    0.0208    0.0126    0.2279    0.2281;
0.1024    0.1227    0.0717    0.2568    0.1071    0.0583    0.1058    0.2568;
0.1384    0.0238    0.1616    0.2106    0.1411    0.1249    0.1585    0.1853;
```

```
    0.1711    0.3254    0.2008    0.1408    0.1358    0.0447    0.3176    0.0559;
    0.1473    0.1395    0.1315    0.1639    0.1332    0.1116    0.1557    0.1587;
    0.2803    0.3254    0.0794    0.0757    0.3128    0.3203    0.1199    0.2454;
    0.3409    0.2090    0.1980    0.2172    0.1526    0.1840    0.2623    0.2201;
    0.1614    0.0176    0.1558    0.2290    0.0410    0.0063    0.1148    0.1354;
    0.1172    0.1193    0.1907    0.0490    0.1304    0.1193    0.2345    0.0671;
    0.0187    0.0277    0.1692    0.2213    0.0395    0.0277    0.1816    0.2213;
    0.0654    0.0238    0.2131    0.0900    0.1688    0.0872    0.2276    0.0341;
    0.1412    0.1400    0.2086    0.1439    0.1586    0.1400    0.1451    0.1372;
    0.0172    0.1529    0.2048    0.1693    0.1364    0.2007    0.1297    0.1483;
    0.2017    0.0819    0.0689    0.2106    0.1656    0.2213    0.1050    0.2123;
    0.3176    0.1414    0.1802    0.2172    0.2280    0.1050    0.2182    0.0972;
    0.0173    0.0126    0.2288    0.1882    0.0439    0.0126    0.1562    0.2454;
    0.0943    0.0726    0.1297    0.2020    0.0885    0.0726    0.1445    0.1729;
    0.0415    0.0208    0.3122    0.2213    0.0222    0.0126    0.2054    0.1909;

P=transpose(Q);
T=[2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ];
T1 = ind2vec(T);
P2=[apna1l apna2l apnd1l apnd2l apna1r apna2r apnd1r apnd2r]
P1=transpose(P2);

disp('spread=0.1')
net = newpnn(P,T1);
 Y = sim(net,P1);
Yc = vec2ind(Y)
ifYc==1
disp('Control Subject')
elseifYc==2
disp('Amyotrophic lateral sclerosis patients')
elseifYc==3
disp('Huntington disease patient')
else
disp('Parkinson Patient')
end
end
end

fprintf('\n'); disp('spread=0.01')
 net1 = newpnn(P,T1,0.01);
 Y1 = sim(net1,P1);
 Yc1 = vec2ind(Y1)

if Yc1==1
disp('Control Subject')
elseif Yc1==2
disp('Amyotrophic lateral sclerosis patients')
elseif Yc1==3
disp('Huntington disease patient')
else
disp('Parkinson Patient')
end
end
end
```

```
fprintf('\n'); disp('spread=0.003')
 net1 = newpnn(P,T1,0.003);
 Y1 = sim(net1,P1);
 Yc1 = vec2ind(Y1)

if Yc1==1
disp('Control Subject')
elseif Yc1==2
disp('Amyotrophic lateral sclerosis patients')
elseif Yc1==3
disp('Huntington disease patient')
else
disp('Parkinson Patient')
end
end
end
```

- **Case-2: Training set=24 and Testing set=20**

```
clc
clearall
loadpark9.ts;
x1=park9(:,2);
x2=park9(:,3);
disp('park9')

sas

Q=[

%0.1834    0.0183    0.2532    0.2795    0.1834    0.2865    0.2022    0.2917;
0.2299    0.2213    0.2141    0.2454    0.2377    0.2213    0.2118    0.2632;
0.0126    0.0178    0.2537    0.2770    0.0126    0.0178    0.2049    0.2770;
0.0998    0.0579    0.2366    0.2370    0.1307    0.1171    0.1517    0.1408;
0.3230    0.1227    0.2543    0.2482    0.3300    0.0195    0.2599    0.2254;
%0.1869    0.2363    0.2705    0.1137    0.1897    0.2865    0.0232    0.2472;
0.0766    0.0631    0.1945    0.2201    0.0629    0.0631    0.2006    0.2068;
0.3299    0.2632    0.1712    0.2123    0.3874    0.1909    0.1524    0.2123;
%0.1886    0.1909    0.1318    0.3168    0.1130    0.1909    0.1215    0.3407;
%0.1023    0.0922    0.1779    0.2148    0.0863    0.0922    0.1779    0.2148;
%0.0935    0.0756    0.1740    0.2463    0.0812    0.0756    0.2595    0.1643;
```

```
%0.2154    0.0816    0.1490    0.1814    0.0650    0.0169    0.1294    0.2318;
0.0560    0.0579    0.1520    0.2221    0.0774    0.0579    0.1299    0.3032;
%0.1278    0.0839    0.1735    0.2201    0.1037    0.0631    0.1361    0.2642;
0.1982    0.0636    0.1352    0.2294    0.1236    0.0232    0.2328    0.1957;
0.0289    0.0126    0.2377    0.2201    0.0289    0.0126    0.1575    0.2290;
0.2207    0.1004    0.1495    0.2213    0.2207    0.0631    0.1131    0.1909;
%0.2648    0.1584    0.1618    0.1303    0.1442    0.0594    0.1709    0.1748;
%0.0660    0.0321    0.1957    0.2159    0.0108    0.0321    0.2043    0.2363;
0.0276    0.0126    0.0948    0.2213    0.0289    0.0126    0.1618    0.1954;
0.0175    0.0207    0.1934    0.1647    0.0175    0.0207    0.1666    0.2366;
%0.0089    0.0149    0.1879    0.3128    0.0404    0.0195    0.1727    0.2880;


%0.1467    0.2272    0.1172    0.2454    0.1972    0.0863    0.1219    0.2272;
0.1088    0.0126    0.2022    0.1978    0.2898    0.1623    0.2362    0.1814;
0.0070    0.0180    0.2252    0.1684    0.0105    0.0126    0.2560    0.1978;
0.0257    0.0232    0.1529    0.2125    0.0257    0.0232    0.1440    0.1814;
%0.1866    0.1321    0.1813    0.2642    0.1380    0.1400    0.2022    0.2172;
%0.1402    0.0126    0.1841    0.1584    0.0687    0.0126    0.1915    0.1951;
%0.0621    0.0232    0.1040    0.1772    0.1658    0.0232    0.0876    0.1772;
0.2742    0.1469    0.1398    0.1807    0.2808    0.1469    0.1445    0.1807;
0.0452    0.0176    0.0673    0.1909    0.0649    0.0176    0.0760    0.1909;
%0.1808    0.1814    0.1802    0.2213    0.1562    0.2106    0.1680    0.2213;
0.3066    0.1584    0.2594    0.2551    0.0208    0.0126    0.2279    0.2281;


%0.1711    0.3254    0.2008    0.1408    0.1358    0.0447    0.3176    0.0559;
%0.1473    0.1395    0.1315    0.1639    0.1332    0.1116    0.1557    0.1587;
0.2803    0.3254    0.0794    0.0757    0.3128    0.3203    0.1199    0.2454;
0.3409    0.2090    0.1980    0.2172    0.1526    0.1840    0.2623    0.2201;
%0.1614    0.0176    0.1558    0.2290    0.0410    0.0063    0.1148    0.1354;
%0.1172    0.1193    0.1907    0.0490    0.1304    0.1193    0.2345    0.0671;
0.0187    0.0277    0.1692    0.2213    0.0395    0.0277    0.1816    0.2213;
%0.1412    0.1400    0.2086    0.1439    0.1586    0.1400    0.1451    0.1372;
0.0172    0.1529    0.2048    0.1693    0.1364    0.2007    0.1297    0.1483;
0.3176    0.1414    0.1802    0.2172    0.2280    0.1050    0.2182    0.0972;
0.0943    0.0726    0.1297    0.2020    0.0885    0.0726    0.1445    0.1729;
];

P=transpose(Q);
T=[2 2 2 2 2 2 3 3 3 3 3 3  1 1 1 1 1 1  4 4 4 4 4 4 ];
T1 = ind2vec(T);
P2=[apna1l apna2l apnd1l apnd2l apna1r apna2r apnd1r apnd2r]
P1=transpose(P2);
disp('spread=0.1')
net = newpnn(P,T1);
 Y = sim(net,P1);
Yc = vec2ind(Y)
ifYc==1
disp('Control Subject')
elseifYc==2
disp('Amyotrophic lateral sclerosis patients')
elseifYc==3
disp('Huntington disease patient')
else
disp('Parkinson Patient')
end
end
end
```

```
fprintf('\n'); disp('spread=0.01')
 net1 = newpnn(P,T1,0.01);
 Y1 = sim(net1,P1);
 Yc1 = vec2ind(Y1)

if Yc1==1
disp('Control Subject')
elseif Yc1==2
disp('Amyotrophic lateral sclerosis patients')
elseif Yc1==3
disp('Huntington disease patient')
else
disp('Parkinson Patient')
end
end
end

fprintf('\n'); disp('spread=0.003')
 net1 = newpnn(P,T1,0.003);
 Y1 = sim(net1,P1);
 Yc1 = vec2ind(Y1)

if Yc1==1
disp('Control Subject')
elseif Yc1==2
disp('Amyotrophic lateral sclerosis patients')
elseif Yc1==3
disp('Huntington disease patient')
else
disp('Parkinson Patient')
end
end
end
```

- In **case 1** all the entropic values of time series data formed the training set Q.
- P is made as transpose of Q matrix.
- T is the corresponding class name for the entropic values in training set.
    - ✓ Class1 represent control subject
    - ✓ Class2 represent Amyotrophic lateral sclerosis patients
    - ✓ Class3 represent Huntington disease patient
    - ✓ Class4 represent Parkinson Patient

  A target matrix with 1s in the desired location is needed. This is possible with the function ind2vec
- A neural network using newpnn (P,T, spread) is created.
- P2 is the testing set and P1 is transpose of it.

- Now the P1 testing set is fed into the neural network and simulate it to get the classified class to which it belongs.

  Y=sim(net,P) Here Y is vector to get the class indices we use vect2ind function.

- In **case 2** training set=24 and testing set=20
  - ✓ Testing set={als2, als4, als5, als6, als8, als10, hunt2, hunt4, hunt6, hunt8, hunt14, hunt15, control2, control5, control6, control11, control12, control14, park3, park4, park7, park10, park12, park14}
  - ✓ Testing set={als1, als7, als11, als12, als13, hunt1, hunt3, hunt12, hunt13, hunt16, control1, control7, control8, control10, control13, park1, park2, park5, park6, park9}

  The entropic values, which were written as comments, form testing set and remaining forms the training set.

Chapter 4

Results and Discussion

A number of graphs were obtained in MATLAB graphical window after feature extraction and processing of gait data which are depicted as follows-

4.1 Graphical window

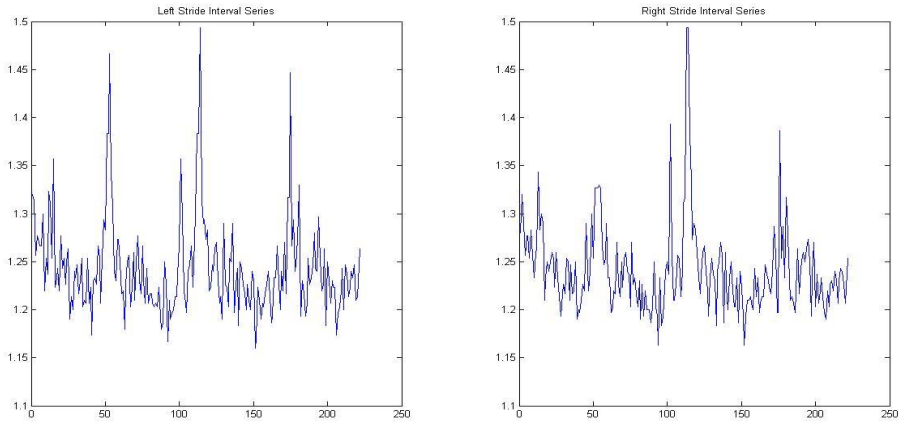

Fig. 4 Stride interval Vs Elapsed time

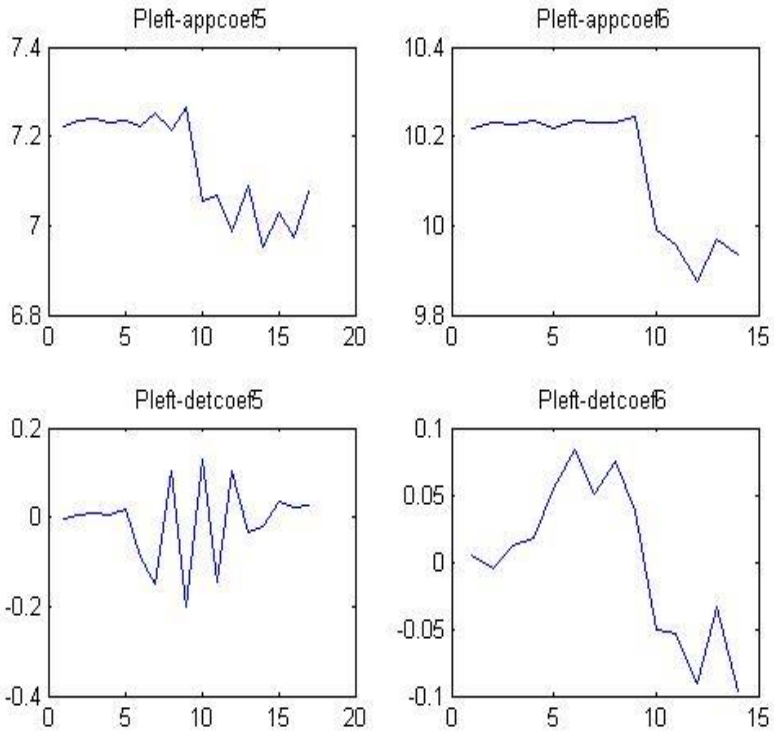

Fig. 5 The coefficients value obtained after wavelet decomposition for left leg

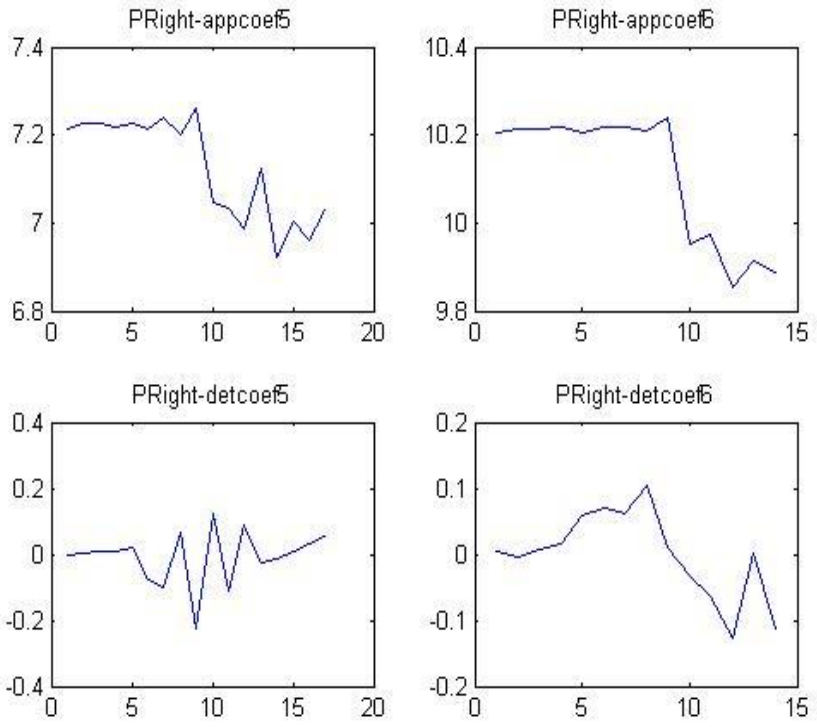

Fig.6 The coefficients value obtained after wavelet decomposition for right leg

- Fig.3 shows relationship between stride intervals and elapsed time in park9.ts. It has been found that the stride intervals lie in the range of 1-1.5 s.
- Fig.4 and Fig.5 simply show the coefficients value obtained after wavelet decomposition. For example Pleft-apcoef5 (A5) is 17*1 vectors which means it has 17 numbers of coefficients each having some coefficient values. So Y axis represents coefficient values and X axis represents coefficient number. Fig.4 shows the graph of coefficient values for left leg whereas Fig.5 shows the graph of coefficient values for left leg
- It is observed that decomposing a wave to higher level reduces the number of coefficients in the respective level
- It can be noted that A6 has 14 number of coefficients as compared to 17 number of coefficients in A5

4.2 Command window

The entropic values of left leg and right leg are obtained of a Parkinson patient and the classification is also achieved for different spread parameters.

4.2.1 Case1 (Training set=63 and Testing set=63)

```
park9
PLeft =
   0.1412    0.1400    0.2086    0.1439
PRight =
   0.1586    0.1400    0.1451    0.1372
P2 =
  0.1412     0.1400     0.2086     0.1439     0.1586     0.1400     0.1451
0.1372

spread =0.1
Yc =
    4
Parkinson Patient

spread=0.01
Yc1 =
    2
Parkinson Patient

Spread =0.003
Yc1 =
    1
Parkinson Patient
>>
```

4.2.2Case1 (Training set=24 and Testing set=20)

```
park9
PLeft =
   0.1412    0.1400    0.2086    0.1439
PRight =
   0.1586    0.1400    0.1451    0.1372
P2 =
  0.1412     0.1400     0.2086     0.1439     0.1586     0.1400     0.1451
0.1372


spread=0.1
Yc =
    4
Parkinson Patient


spread=0.01
Yc1 =
    2
Amyotrophic lateral sclerosis patients


spread=0.003
Yc1 =
    1
Control Subject
>>
```

- It has been observed that for spread 0.1, 0.01 and 0.003 the obtained class index matches with original Parkinson class(4) in case 1
- But in case 2 it is wrongly classified as class 2 for spread 0.01 and class 1 for a spread 0.003

4.3 Datasheet for output classified class (case-1)

The entropic values of all the 63 patients were taken as a training set for the neural network and classification was obtained for different spread parameters. The output is shown in the tabular form:

	spread 0.1	spread 0.01	spread 0.003
als1	2	2	2
als2	2	2	2
als3	2	2	2
als4	3	3	3
als5	2	2	2
als6	2	2	2
als7	2	2	2
als8	3	2	2
als9	2	2	2
als10	2	2	2
als11	2	2	2
als12	3	2	2
als13	3	2	2
hunt1	3	3	3
hunt2	3	3	3
hunt3	3	3	3
hunt4	3	3	3
hunt5	3	3	3
hunt6	3	3	3
hunt7	3	3	3
hunt8	3	3	3
hunt9	4	3	3
hunt10	3	3	3
hunt11	3	3	3
hunt12	3	3	3

hunt13	3	3	3
hunt14	3	3	3
hunt15	3	3	3
hunt16	3	3	3
hunt17	3	3	3
hunt18	3	3	3
hunt19	3	3	3
control1	1	1	1
control2	1	1	1
control3	1	1	1
control4	1	1	1
control5	3	1	1
control6	3	1	1
control7	1	1	1
control8	3	1	1
control9	1	1	1
control10	1	1	1
control11	1	1	1
control12	1	1	1
control13	1	1	1
control14	1	1	1
control15	1	1	1
control16	1	1	1
park1	4	4	4
park2	4	4	4
park3	4	4	4
park4	4	4	4
park5	4	4	4
park6	4	4	4
park7	3	4	4
park8	4	4	4
park9	4	4	4
park10	4	4	4
park11	1	4	4
park12	4	4	4
park13	3	4	4
park14	1	4	4
park15	4	4	4

2. Datasheet for output classified class (case-1)

4.4 Datasheet for output classified class (case-2)

The entropic values of all the 24 patients were taken as a training set and the values of 20 patients formed the testing set for the neural network and classification was obtained for different spread parameters. The output is shown in the tabular form:

	spread 0.1	spread 0.01	spread 0.003
als1	2	2	1
als7	2	2	1
als11	2	2	1
als12	2	2	2
als13	2	2	2
hunt1	3	3	1
hunt3	3	3	3
hunt12	3	3	1
hunt13	3	4	4
hunt16	3	2	2
control1	3	3	1
control7	2	3	1
control8	3	2	1
control10	1	1	1
control13	2	2	1
park1	4	4	1
park2	4	4	1
park5	3	4	1
park6	4	4	1
park9	4	2	1

3. Datasheet for output classified class (case-2)

Chapter 5

Diagnoses Using Probabilistic Neural Network

The reliability of a diagnosis test can be measured based upon several attributes:-

- Sensitivity
- Specificity
- Accuracy

To calculate these attributes, these parameters are used:-

- **True Positive(TP)**:- If a disease is proven present in a patient and the diagnosis tests also indicates the presence of the disease the test is considered to be true positive
- **True Negative(TN)**:- Patient is not suffering from the disease and the diagnosis test also proves disease not being present
- **False Positive(FP)**:- Patient is not suffering from the disease but the diagnosis test proves disease being present
- **False Negatives(FN)**:- Patient is suffering from the disease but the diagnosis test proves disease not being present

Number of TP, TN, FP, FN for each disease is observed and is listed in a tabular form for both the cases

5.1 Number of TP, TN, FP, and FN (case-1)

spread 0.1	als	9	50	0	4
	control	13	45	2	3
	hunt	18	35	9	1
	park	11	47	1	4
spread 0.01	als	12	50	0	1
	control	16	47	0	0
	hunt	19	43	1	0
	park	15	48	0	0
spread 0.003	als	12	50	0	1
	control	16	47	0	0
	hunt	19	43	1	0
	park	15	48	0	0

4. Number of TP, TN, FP, and FN (case-1)

5.2 Number of TP, TN, FP, and FN (case-2)

		TP	TN	FP	FN
spread 0.1	als	5	13	2	0
	control	1	15	0	4
	hunt	5	12	3	0
	park	4	15	0	1
spread 0.01	als	5	11	4	0
	control	1	15	0	4
	hunt	3	13	2	2
	park	4	14	1	1
spread 0.003	als	2	14	1	3
	control	5	5	10	0
	hunt	1	15	0	4
	park	0	14	1	5

5. Number of TP, TN, FP, and FN (case-2)

Now using these terms the sensitivity, specificity and accuracy of the diagnosis are measured.

- **Sensitivity**- It detects the efficiency of a test in detecting a positive disease. Sensitivity is equal to TP/(TP+TN)
- **Specificity**- It estimates how likely patients without the diseases can be correctly ruled out. Specificity is equal to TN/(TN+TP)
- **Accuracy**- It measures how correctly a diagnostic test identifies and excludes a given condition. Accuracy is equal to(TN+TP)/(TN+TP+FP+FN)

The measured sensitivity, specificity, and accuracy is depicted in tabular form for both the cases

5.3Sensitivity, Specificity and Accuracy of the diagnosis (Case 1)

	Accuracy=(TN+TP)/(TN+TP+FP+FN)		
	spread 0.1	spread 0.01	spread 0.003
Als	93.65%	98.41%	98.41%
Control	92.06%	100%	100%
Hunt	84.13%	98.41%	98.41%
Park	92.06%	100%	100%
	Specificity=TN/(TN+FP)		
	spread 0.1	spread 0.01	spread 0.003
Als	100%	100%	100%
Control	95.74%	100%	100%
Hunt	79.55%	97.72%	97.72%
Park	97.92%	100%	100%
	Sensitivity=TP/(TP+FN)		
	spread 0.1	spread 0.01	spread 0.003
Als	69.23%	92.30%	92.30%
Control	81.25%	100%	100%
Hunt	94.73%	100%	100%
Park	73.33%	100%	100%

6. Sensitivity, Specificity and Accuracy of the diagnosis (Case 1)

5.4 Sensitivity, Specificity and Accuracy of the diagnosis (Case 2)

	Accuracy=(TN+TP)/(TN+TP+FP+FN)		
	spread 0.1	spread 0.01	spread 0.003
Als	90%	80%	80%
Control	80%	80%	50%
Hunt	85%	80%	80%
Park	95%	90%	70%
	Specificity=TN/(TN+FP)		
	spread 0.1	spread 0.01	spread 0.003
Als	86.67%	73.33%	93.33%
Control	100%	100%	33.33%
Hunt	80%	86.67%	100%
Park	100%	93.33%	93.33%
	Sensitivity=TP/(TP+FN)		
	spread 0.1	spread 0.01	spread 0.003
Als	100.00%	100.00%	40.00%
Control	20%	20%	100.00%
Hunt	100%	60.00%	20%
Park	80%	80.00%	0.00%

7. Sensitivity, Specificity and Accuracy of the diagnosis (Case 2)

Conclusion

In this thesis a diagnosis method (neural network) was created for patients with neurodegenerative disorders by taking their gait analysis data. It was found that the probabilistic neural network classifies the class of diseases with a much higher accuracy in comparison to other methods such as linear discriminant analysis, quadratic discriminant analysis, logistic classifier, fisher's classifier, decision tree classifier, support vector machine classifier, etc. [17]. This non-invasive approach would help the physicians to identify the disease and hence would help in the correct treatment of the patients. When a training set of 24 and a testing set of 20 was taken, the accuracy of classification of patients obtained were as below:-

Amyotropic lateral sclerosis patients-90%

Control subjects-80%

Huntington patients-85%

Parkinson patients-95%

Bibliography

1. Bennett DA, Beckett LA, Murray AM et al. *Prevalence of parkinsonian signs and associated mortality in a community population of older people*. N Engl J Med 1996; 334:71-76
2. Masood Banaiea, Mohammad Pooyanb, Mohammad Mikailib *Introduction and application of an automatic gait recognition method to diagnose movement disorders that arose of similar causes* Expert Systems with Applications 38(6) June 2011, Pages 7359–7363
3. Anthony E. Lang, M.D., and Andres M. Lozano, M.D. N Engl J Med. *Parkinson's Disease* 1998; 339:1044-1053
4. M. Sifuzzaman1, M.R. Islam and M.Z. Ali. *Application of Wavelet Transform and its Advantages Compared to Fourier Transform*. Journal of Physical Sciences, Vol. 13, 2009, 121-134
5. X.Q. Jiang et al. *Application of the lifting wavelet to rough surfaces* Precision Engineering, vol. 25, 2001, pp. 83−89
6. Pankaj Sapra, Rupinderpal Singh, Shivani Khurana. *Brain Tumor Detection Using Neural Network* Volume-1, Issue-9. International Journal of Science and Modern Engineering (IJISME) August 2013 ISSN: 2319-6386.
7. Pincus, S. M.; Gladstone, I. M.; Ehrenkranz, R. A. (1991). *A Regularity Statistic for Medical Data Analysis* **7** (4): 335–345.
8. Pincus SM. *Approximate entropy as a measure of system complexity* *Proc Natl Acad Sci USA* 1991; 88:2297-2301.
9. Richman JS, Moorman JR. *Physiological time-series analysis using approximate entropy and sample entropy* *Am J Physiol Heart Circ Physiol* , 2000 **278**(6): 2039-2049

10. Mantzaris, D.H., Alexandroupolis; Anastassopoulos, G.C.; Lymberopoulos, D.K. Medical disease prediction using Artificial Neural Networks. *Bioinformatics and Bioengineering, 2008*. (BIBE 2008). 8th IEEE International Conference 2008 pg: 1-6
11. Masood Banaiea , Mohammad Pooyanb, Mohammad Mikaili Introduction and application of an automatic gait recognition method to diagnose movement disorders that arose of similar causes, Expert Systems with Applications, 38(6) 2011, Pages 7359–7363
12. *Quantification of human motion: gait analysis—benefits and limitations to its application to clinical problems* Journal of Biomechanics Volume 37, Issue 12, 2004, Pages 1869–1880
13. Weijun Tao 1, Tao Liu 2,Rencheng Zheng 3 and Hutian Feng *Gait Analysis Using Wearable Sensors* 2012
14. *A comparison of multiple classification methods for diagnosis of Parkinson disease* Expert Systems with Applications Volume 37, Issue 2, 2010, Pages 1568–1572
15. Wei-Hsin Wang Yu-Liang Hsu ; Ming-Chyi Pai ; Cheng-Hsiung Wang ; Chun-Yao Wang ; Chien-Wen Lin ; Hao-Li Wu ; Pau-Choo Chung International Joint Conference on Neural Networks(IJCNN), 2014; 3251 – 3257
16. Shouman, Turner, T. ; Stocker, R. Electronics, Communications and Computers (JEC-ECC), 2012 Japan-Egypt Conference, 2012 pg: 173 – 176
17. Shamaila Iram, Dhiya Al-Jumeily, Paul Fergus, Martin Randles, Abir Hussain, *Computational Data Analysis for Movement Signals Based on Statistical Pattern Recognition Techniques for Neurodegenerative Diseases*