A

Thesis

On

CLASSIFICATION OF ELECTROCARDIOGRAM WAVEFORMS USING PNN

Submitted by

Swaraj Mohapatra

B.Tech, Final Year

Roll no-10600025

Under the guidance of

Dr. Madhusree kundu

Department of Chemical Engineering

NIT Rourkela, Orissa

CLASSIFICATION OF ELECTROCARDIOGRAM WAVEFORMS USING PNN

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

Bachelor of Technology
In
Chemical Engineering

By

Swaraj Mohapatra

Under the supervision of
Dr. Madhusree Kundu
Chemical Engineering
NIT Rourkela

National Institute Of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, ***"Classification of ECG signals using PNN"*** by Swaraj Mohapatra in partial requirements for the curriculum requirement of Bachelor of technology in Chemical Engineering at National Institute of Technology, Rourkela, is an authentic work carried out by them under my supervision and guidance. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree.

Dr. Madhusree Kundu
Chemical Engineering NIT,Rourkela

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my teacher and guide, Dr. Madhusree Kundu who infused me with the drive to carry on the project work by her consistent support and encouragement throughout the project work. Without her support, the thesis would have never been completed. I would like to thank the faculty of Chemical Engineering department of NIT, Rourkela for being the panacea to all my queries. Last but not the least, I would like to thank all my dear friends for being their wikipediaic knowledge and unrelenting support.

Regards,

Swaraj Mohapatra

CONTENTS

.

ABSTRACT

With increasing computational power, sophisticated algorithms have been proposed to improve the prediction accuracy of ECG waveform classification systems. One such approach is application of PCA on the features extracted using curve fitting tool in MATLAB and application of PNN to get the classification desired for various purposes. The present work reviews the methods of feature extraction and then the application of PNN to get the classification. It also reveals an approach that can be taken for ECG waveform classification based works

NOMENCLATURE

X^T	Data Matrix
Y^T	Matrix In Which Each Row Of Y^T Is A Rotation Of The Corresponding Row Of X^T
W	Orthogonal Matrix
Σ	Diagonal Matrix
V	Matrix
L	Singular Vectors
C	Observer Covariances
K	No. of Classes
N	No. of Nodes
x	Input Feature Vector
F	Spread Parameters (Standard Deviations)
N	Dimension of The Input Vectors
P	Number of Center Vectors In Class 1
R	Number of Center Vectors In Class 2

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION:

The recognition of the ECG beats is a very important task in the coronary intensive unit, where the classification of the ECG beats is essential tool for the diagnosis. Up to now, many algorithms have been developed for the recognition and classification of ECG signal. Some of them use either time or frequency domain representation, on the basis of which many specific attributes are defined, allowing the recognition between the beats belonging to different pathological classes. The ECG waveforms may differ for the same patient to such extent that they are unlike each other and at the same time alike for different types of beats (Osowski and Linh, 2001). Artificial neural network and fuzzy-based techniques were also employed to exploit their natural ability in pattern recognition task for successful classification of ECG beats (Hu et al., 1997). Analysis of the ECG signals is of the great importance in the detection of cardiac anomalies. One of the most important ECG components is the QRS complex, which is associated with electrical ventricular activation (Barro et al., 1998). ECG pattern recognition can be divided into a sequence of stages; starting with feature extraction from the occurring patterns, which is the conversion of the patterns to features that are regarded as a condensed representation. In the next step, the feature selection, smaller number of meaningful features, that the best represents the given pattern without redundancy, is defined. Finally, the classification is carried out, i.e., specific pattern is assigned to a specific class according to the characteristic features selected for it (Dickhaus and Heinrich, 1996).

BACKGROUND

1.2 The Electrocardiogram:

Electrocardiography (ECG) is a transthoracic interpretation of the electrical activity of the heart over time captured and externally recorded by skin electrodes. [ECG simplified by Ashwin Kumar]. The ECG works by detecting and amplifying the tiny electrical changes on the skin that are caused when the heart muscle "depolarises" during each heartbeat. At rest, each heart muscle cell has a charge across its outer wall, or cell membrane. Reducing this charge towards zero is called de-polarisation, which activates the mechanisms in the cell that cause it to contract. During each heartbeat a healthy heart will have an orderly progression of a wave of depolarisation that is triggered by the cells in the sinoatrial node, spreads out through the atrium, passes through "intrinsic conduction pathways" and then spreads all over the ventricles. This is detected as tiny rises and falls in the voltage between two electrodes placed either side of the heart which is displayed as a wavy line either on a screen or on paper. This display indicates the overall rhythm of the heart and weaknesses in different parts of the heart muscle.

Usually more than 2 electrodes are used and they can be combined into a number of pairs. (For example: Left arm (LA), right arm (RA) and left leg (LL) electrodes form the pairs: LA+RA, LA+LL, RA+LL) The output from each pair is known as a lead. Each lead is said to look at the heart from a different angle. Different types of ECGs can be referred to by the number of leads that are recorded, for example 3-lead, 5-lead or 12-lead ECGs (sometimes simply "a 12-lead"). A 12-lead ECG is one in which 12 different electrical signals are recorded at approximately the same time and will often be used as a one-off recording of an ECG, typically printed out as a paper copy. 3- and 5-lead ECGs tend to be monitored continuously and viewed only on the screen of an appropriate monitoring device, for example during an operation or whilst being transported in an ambulance. There may, or may not be any permanent record of a 3- or 5-lead ECG depending on the equipment used. The ECG waveform is shown in the figure 2.1here.

ECG WAVEFORM:

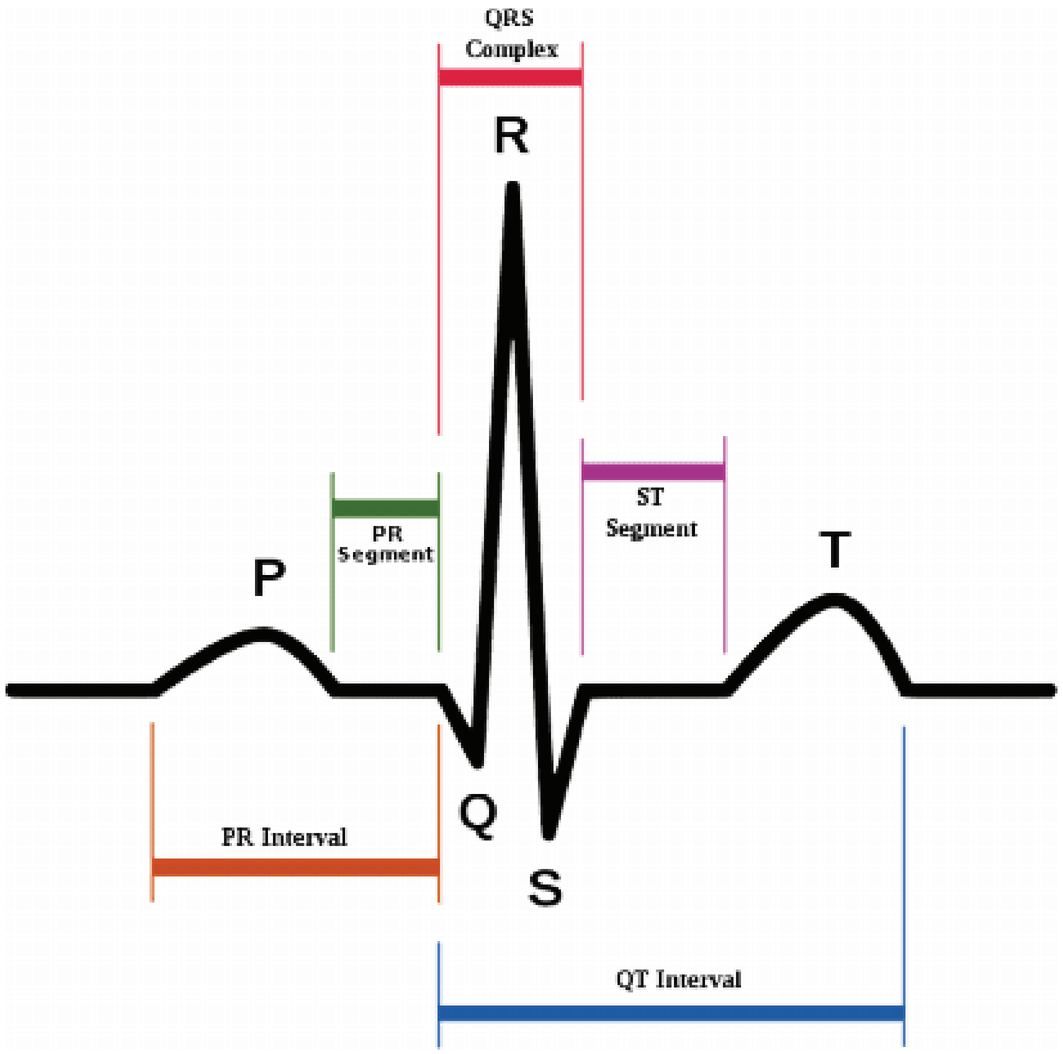

Figure 1.1: ECG Waveform

The ECG waveform can be broken down into three important parts each denoting a peak on the either side represented by P, Q, R, S, T. each of them represent a vital processes in the heart and those processes have been illustrated in table 2.1. In case of a disease afflicting the heart, the waves get distorted according to the area which is not functioning normally. Thus by inspection of the ECG waveform the nature of disease can be found out easily.

Table 2.1: The Description And Duration Of Each Wave In The ECG Waveform.

Feature	**Description**	**Duration**
RR interval	The interval between an R wave and the next R wave is the inverse of the heart rate. Normal resting heart rate is between 50 and 100 bpm	0.6 to 1.2s
P wave	During normal atrial depolarization, the main electrical vector is directed from the SA node towards the AV node, and spreads from the right atrium to the left atrium. This turns into the P wave on the ECG	80ms
PR interval	The PR interval is measured from the beginning of the P wave to the beginning of the QRS complex. The PR interval reflects the time the electrical impulse takes to travel from the sinus node through the AV node and entering the ventricles. The PR interval is therefore a good estimate of AV node function.	120 to 200ms
PR segment	The PR segment connects the P wave and QRS complex. This coincides with the electrical conduction from the AV node to the bundle of His to the bundle branches and then to the Purkinje Fibers. This electrical activity does not produce a contraction directly and is merely travelling down towards the ventricles and this shows up flat on the ECG. The PR interval is more clinically relevant.	50 to 120ms
QRS complex	The QRS complex reflects the rapid depolarization of the right and left ventricles. They have a large muscle mass compared to the atria and so the QRS complex usually has a much larger amplitude than the P-wave.	80 to 120ms
J-point	The point at which the QRS complex finishes and the ST segment begins. Used to measure the degree of ST elevation or depression present.	N/A

Table 2.1: The Description And Duration Of Each Wave In The ECG Waveform.

Feature	**Description**	**Duration**
ST segment	The ST segment connects the QRS complex and the T wave. The ST segment represents the period when the ventricles are depolarized. It is isoelectric.	80 to 100ms
T wave	The T wave represents the repolarization (or recovery) of the ventricles. The interval from the beginning of the QRS complex to the apex of the T wave is referred to as *absolute refractory period.* The last half of the T wave is referred to as the *relative refractory period* (or vulnerable period).	160ms
ST interval	The ST interval is measured from the J point to the end of the T wave.	320ms
QT interval	The QT interval is measured from the beginning of the QRS complex to the end of the T wave. A prolonged QT interval is a risk factor for ventricular tachyarrhythmia and sudden death. It varies with heart rate and for clinical relevance requires a correction for this, giving the QTc.	300 to 430ms
U wave	The U wave is not always seen. It is typically low amplitude, and, by definition, follows the T wave.	

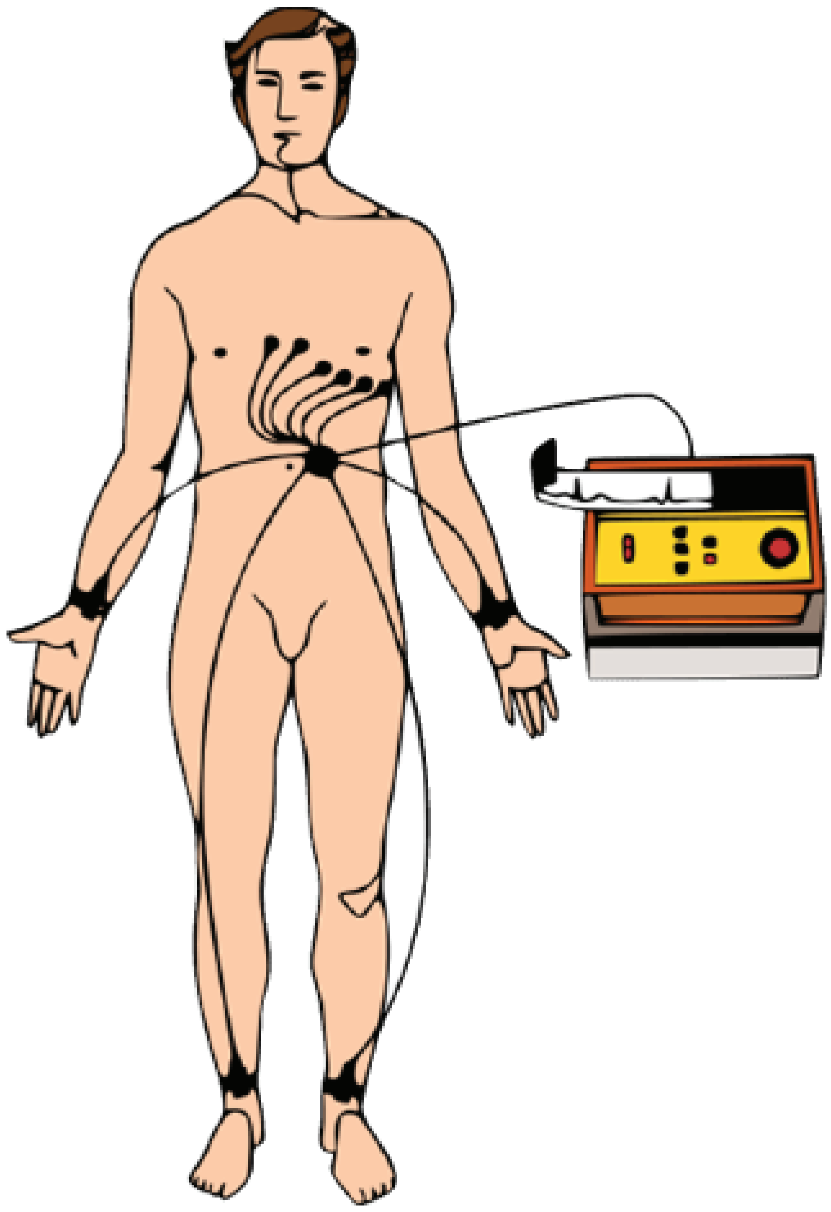

Figure 1.2: Image Showing A Patient Connected To The 10 Electrodes Necessary For A 12-Lead ECG.

CHAPTER-2

METHODS

For efficient classification of the ECG waveforms, proper feature extraction has to be done. The methods and materials used for this purpose are as follows:

1. Curve Splitting Tool [MATLAB]
2. Baseline wandering using polynomial cubic spline.
3. Normalization
4. PCA [Principal Component Analysis]
5. PNN

2.1 Curve Splitting Tool [MATLAB]:

Curve fitting methods allow you to create, access, and modify curve fitting objects. They also allow you, through methods like plot and integrate, to perform operations that uniformly process the entirety of information encapsulated in a curve fitting object. Curve Fitting Toolbox software provides a variety of methods for data analysis and modeling. In application, these methods are applied in a systematic manner, which can be represented in a standard workflow diagram such as the one below.

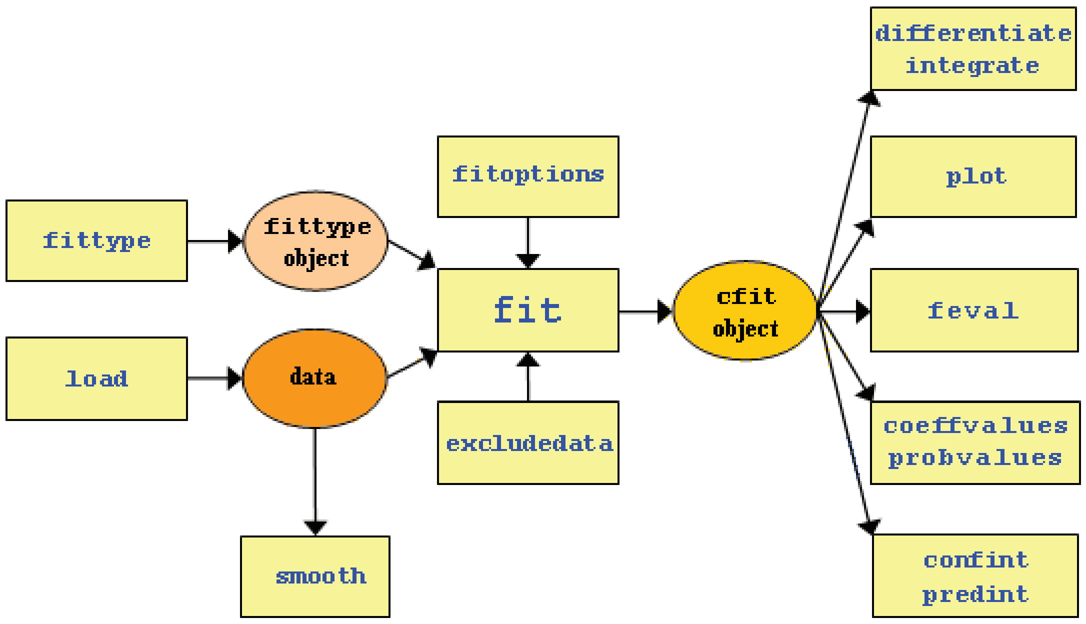

Fig 2.1: Workflow Diagram Of cfit Tool

2.1.1 A typical analysis using curve fitting methods proceeds as follows:

1. Import your data into the MATLAB workspace using the load command (if your data has previously been stored in MATLAB variables) or any of the more specialized MATLAB functions for reading data from particular file types.
2. If your data is noisy, you might want to smooth it using the smooth function. Smoothing is used to identify major trends in the data that can assist you in choosing an appropriate

family of parametric models. If a parametric model is not evident or appropriate, smoothing can be an end in itself, providing a nonparametric fit of the data.

3. A parametric model for the data—either a Curve Fitting Toolbox library model or a custom model that you define—is specified as a fittype object using the fittype function. Library models are displayed with the cflibhelp function.
4. A fit options structure can be created for the fit using the fitoptions function. Fit options specify things like weights for the data, fitting methods, and low-level options for the fitting algorithm.
5. An exclusion rule can be created for the fit using the excludedata function. Exclusion rules indicate which data values will be treated as outliers and excluded from the fit.
6. Data, a fittype object, and (optionally) a fit options structure and an exclusion rule are all passed to the fit function to perform the fit. The fit function returns a cfit object that encapsulates the computed coefficients and the fit statistics.
7. cfit objects returned by the fit function can then be passed to a variety postprocessing functions, such as feval, differentiate, integrate, plot, coeffvalues, probvalues, confint, and predint.

2.2 BASELINE WANDERING:

It is mainly done to remove the noises from the ECG waveforms. The ECG waveforms we get contain noises due to various processes in the body, such as respiration, muscle expansion , muscle contraction etc. To remove these noises baseline wandering is done.

It helps in extracting accurate features from the waveform. And it also helps in the calculation of it.

2.3 NORMALIZATION:

Normalization is the process of isolating statistical error in repeated measured data. A normalization is sometimes based on a property. Quantile normalization, for instance, is normalization based on the magnitude (quantile) of the measures. In another usage in statistics, normalization refers to the division of multiple sets of data by a common variable in order to negate that variable's effect on the data, thus allowing underlying characteristics of the data sets to be compared: this allows data on different scales to be compared, by bringing them to a common scale. In terms of levels of measurement, these ratios only make sense for ratio measurements (where ratios of measurements are meaningful), not interval measurements (where only distances are meaningful, but not ratios)

It is done to normalize the data extracted from the waveforms. As, the features extracted are of different dimensions, so to prevent the higher dimension data from getting the importance, normalization is done.

2.4 PRINCIPAL COMPONENT ANALYSIS [PCA]:

Principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a number of uncorrelated variables called principal components, related to the original variables by an orthogonal transformation. This transformation is defined in such a way that the first principal component has as high a variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components. PCA is sensitive to the relative scaling of the original variables. Depending on the field of application, it is also named the discrete Karhunen–Loève transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD).

PCA was invented in 1901 by Karl Pearson. Now it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA can be done by eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The results of a PCA are usually discussed in terms of component scores (the transformed variable values corresponding to a particular case in the data) and loadings (the variance each original variable would have if the data were projected onto a given PCA axis) (Shaw, 2003).

PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense) most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

PCA is closely related to factor analysis; indeed, some statistical packages (such as Stata) deliberately conflate the two techniques. True factor analysis makes different assumptions about the underlying structure and solves eigenvectors of a slightly different matrix.

2.4.1 Details:

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Define a data matrix, X^T, with zero empirical mean (the empirical mean of the distribution has been subtracted from the data set), where each of the n rows represents a different repetition of the experiment, and each of the m columns gives a particular kind of datum (say, the results from a particular probe). (Note that what we are calling X^T is often alternatively denoted as X itself.) The PCA transformation is then given by:

$$\begin{aligned}\mathbf{Y}^{\mathrm{T}} &= \mathbf{X}^{\mathrm{T}}\mathbf{W} \\ &= \mathbf{V}\boldsymbol{\Sigma}^{\mathrm{T}}\end{aligned} \tag{2.1}$$

where the matrices W, Σ, and V are given by a singular value decomposition (SVD) of X as W Σ V^T. (V is not uniquely defined in the usual case when m<n−1, but Y will usually still be uniquely defined.) Σ is an m-by-n diagonal matrix with nonnegative real numbers on the diagonal. Since W (by definition of the SVD of a real matrix) is an orthogonal matrix, each row of Y^T is simply a rotation of the corresponding row of X^T. The first column of Y^T is made up of the "scores" of the cases with respect to the "principal" component, the next column has the scores with respect to the "second principal" component, and so on.

If we want a reduced-dimensionality representation, we can project X down into the reduced space defined by only the first L singular vectors, WL:

$$\mathbf{Y} = \mathbf{W_L}^T\mathbf{X} = \boldsymbol{\Sigma}_\mathbf{L}\mathbf{V_L}^T \tag{2.2}$$

The matrix W of singular vectors of X is equivalently the matrix W of eigenvectors of the matrix of observed covariances $C = X X^T$,

$$XX^T = W\Sigma\Sigma^T W^T \tag{2.3}$$

Given a set of points in Euclidean space, the first principal component corresponds to a line that passes through the multidimensional mean and minimizes the sum of squares of the distances of the points from the line. The second principal component corresponds to the same concept after all correlation with the first principal component has been subtracted out from the points. The singular values (in Σ) are the square roots of the eigenvalues of the matrix XX^T. Each eigenvalue is proportional to the portion of the "variance" (more correctly of the sum of the squared distances of the points from their multidimensional mean) that is correlated with each eigenvector. The sum of all the eigenvalues is equal to the sum of the squared distances of the points from their multidimensional mean. PCA essentially rotates the set of points around their mean in order to align with the principal components. This moves as much of the variance as possible (using an orthogonal transformation) into the first few dimensions. The values in the remaining dimensions, therefore, tend to be small and may be dropped with minimal loss of information. PCA is often used in this manner for dimensionality reduction. PCA has the distinction of being the optimal orthogonal transformation for keeping the subspace that has largest "variance" (as defined above). This advantage, however, comes at the price of greater computational requirements if compared, for example and when applicable, to the discrete cosine transform. Nonlinear dimensionality reduction techniques tend to be more computationally demanding than PCA.

PCA is sensitive to the scaling of the variables. If we have just two variables and they have the same sample variance and are positively correlated, then the PCA will entail a rotation by 45° and the "loadings" for the two variables with respect to the principal component will be equal. But if we multiply all values of the first variable by 100, then the principal component will be almost the same as that variable, with a small contribution from the other variable, whereas the second component will be almost aligned with the second original variable. This means that whenever the different variables have different units (like temperature and mass), PCA is a somewhat arbitrary method of analysis. (Different results would be obtained if one used Fahrenheit rather than Celsius for example.) Note that Pearson's original paper was entitled "On Lines and Planes of Closest Fit to Systems of Points in Space" – "in space" implies physical Euclidean space where such concerns do not arise. One way of making the PCA less arbitrary is to use variables scaled so as to have unit variance.

2.5 PROBABILISTIC NEURAL NETWORKS [PNN]:

Probabilistic neural networks are forward feed networks built with three layers. They are derived from Bayes Decision Networks. They train quickly since the training is done in one pass of each training vector, rather than several. Probabilistic neural networks estimate the probability density function for each class based on the training samples.

The probabilistic neural network uses Parzen or a similar probability density function. This is calculated for each test vector. This is what is used in the dot product against the input vector as described below. Usually a spherical Gaussian basis function is used, although many other functions work equally well.

Vectors must be normalized prior to input into the network. There is an input unit for each dimension in the vector. The input layer is fully connected to the hidden layer. The hidden layer has a node for each classification. Each hidden node calculates the dot product of the input vector with a test vector subtracts 1 from it and divides the result by the standard deviation squared. The output layer has a node for each pattern classification. The sum for each hidden node is sent to the output layer and the highest values wins.

The Probabilistic neural network trains immediately but execution time is slow and it requires a large amount of space in memory. It really only works for classifying data. The training set must be a thorough representation of the data. Probabilistic neural networks handle data that has spikes and points outside the norm better than other neural nets.

2.5.1 The Architecture of Probabilistic Neural Networks:

A probabilistic neural network (PNN) has 3 layers of nodes. The figure below displays the architecture for a PNN that recognizes K = 2 classes, but it can be extended to any number K of classes. The input layer (on the left) contains N nodes: one for each of the N input features of a feature vector. These are fan-out nodes that branch at each feature input node to all nodes in the hidden (or middle) layer so that each hidden node receives the complete input feature vector x.

The hidden nodes are collected into groups: one group for each of the K classes as shown in the figure. Its associated feature vector in the kth class (there is a Gaussian for each exemplar feature vector). All of the Gaussians in a class group feed their functional values to the same output layer node for that class, so there are K output nodes.

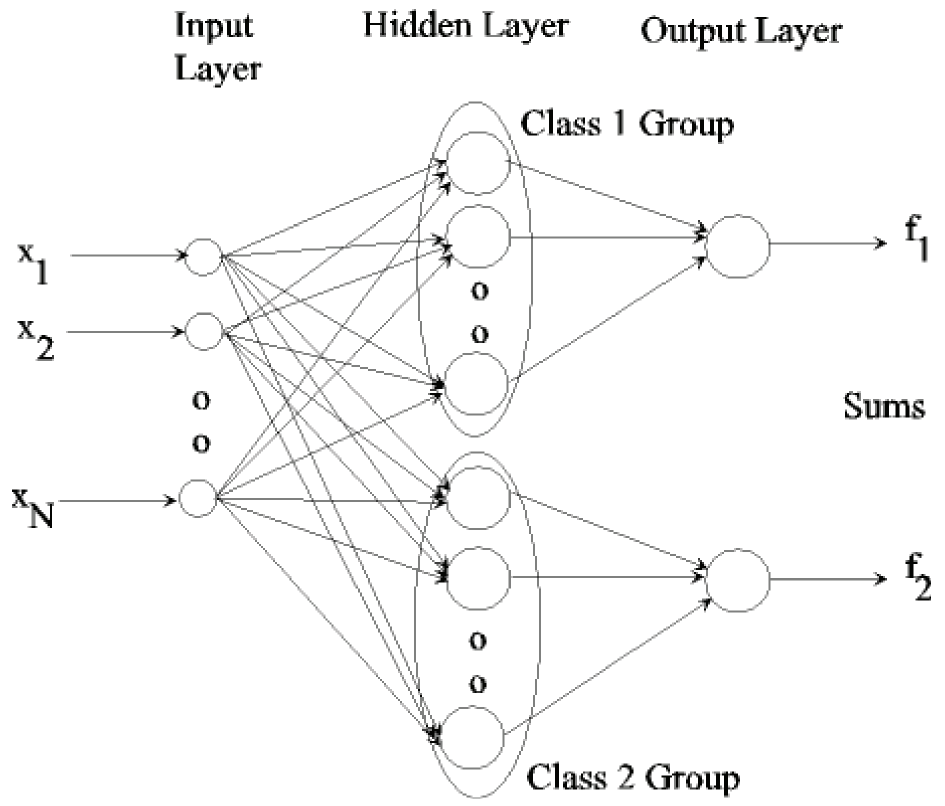

Fig 2.2: Collected Group Of Hidden Nodes

2.5.2 How the PNN Works:

At the output node for Class k (k = 1 or 2 here), all of the Gaussian values for Class k are summed and the sum is scaled to so the probability volume under the sum function is unity so that the sum forms a probability density function. Here we temporarily use special notation for clarity. Let there be P exemplar feature vectors {x(p): p = 1,...,P} labeled as Class 1 and let there be Q exemplar feature vectors {y(r): r = 1,...,R} labeled as Class 2. In the hidden layer there are P nodes in the group for Class 1 and R nodes in the group for Class 2. The equations for each Gaussian centered on the respective Class 1 and Class 2 points x(p) and y(q) (feature vectors) are (where N is the dimension of the vectors) are, for any input vector x

$$g_1(\mathbf{x}) = [1/\sqrt{(2\pi\sigma^2)^N}]\exp\{-\|\mathbf{x} - \mathbf{x}^{(p)}\|^2/(2\sigma^2)\} \quad 2.4$$

$$g_2(\mathbf{y}) = [1/\sqrt{(2\pi\sigma^2)^N}]\exp\{-\|\mathbf{y} - \mathbf{y}^{(q)}\|^2/(2\sigma^2)\} \quad 2.5$$

The F values can be taken to be one-half the average distance between the feature vectors in the same group or at each exemplar it can be one-half the distance from the exemplar to its nearest other exemplar vector. The kth output node sums the values received from the hidden nodes in thekth group, called mixed Gaussians or Parzen windows. The sums are defined by

$$f_1(\mathbf{x}) = [1/\sqrt{(2\pi\sigma^2)^N}](1/P)\Sigma_{(p=1,P)} \exp\{-\|\mathbf{x} - \mathbf{x}^{(p)}\|^2/(2\sigma^2)\} \quad 2.6$$

$$f_2(\mathbf{y}) = [1/\sqrt{(2\pi\sigma^2)^N}](1/Q)\Sigma_{(q=1,Q)} \exp\{-\|\mathbf{y} - \mathbf{y}^{(q)}\|^2/(2\sigma^2)\} \quad 2.7$$

where x is any input feature vector, F1 and F2 are the spread parameters (standard deviations) for Gaussians in Classes 1 and 2 , respectively, N is the dimension of the input vectors, P is the number of center vectors in Class 1 and R is the number of centers in Class 2, x(p) and y(r) are centers in the respective Classes 1 and 2, and 2x - x(p)2 is the Euclidean distance (square root of the sum of squared differences) between x and x(p). Any input vector x is put through both sum functions f1(x) and f2(x) and the maximum value (maximum a posteriori, or MAP value) of f1(x) and f2(x) decides the class. For K > 2 classes the process is analogous. There is no iteration nor computation of weights. For a large number of Gaussians in a sum, the error buildup can be significant. Thus the feature vectors in each class may be reduced by thinning those that are too close to another one and making F larger.

2.5.3 A High Level Algorithm:

We are given the exemplar feature vectors that make up the training set. For each one we know the class to which it belongs. The following sets up the PNN.

Step 1.Read in the file of exemplar vectors and class numbers.

Step 2.Sort these into the K sets where each set contains one class of vectors.

Step 3.For each k define a Gaussian function centered on each exemplar vector in set k define the summed Gaussian output function.

Once the PNN is defined, then we can feed vectors into it and classify them as follows.

Step 1.Read input vector and feed it to each Gaussian function in each class

Step 2.For each group of hidden nodes, compute all Gaussian functional values at the hidden nodes

Step 3.For each group of hidden nodes, feed all its Gaussian functional values to the single output node for that group.

Step 4.At each class output node, sum all of the inputs and multiply by constant

Step 5.Find maximum value of all summed functional values at the output nodes

2.5.4 Syntax:

net = newpnn(P,T,spread)

2.5.5 Description:

Probabilistic neural networks (PNN) are a kind of radial basis network suitable for classification problems.

net = newpnn(P,T,spread) takes two or three arguments,

PR x Q matrix of Q input vectorsTS x Q matrix of Q target class vectorsspreadSpread of radial basis functions (default = 0.1) and returns a new probabilistic neural network. If spread is near zero, the network acts as a nearest neighbor classifier. As spread becomes larger, the designed network takes into account several nearby design vectors.

CHAPTER-3

RESULTS AND DISCUSSIONS

3.1 Baseline wandering and its result:

The main aim was to format the ECG waveforms in such a way that it should give accurate features.as, the ECG taken contained lots of noise, the noise was first removed using the curve fitting tool available in the MATLAB. The ECG waveform with noise is shown in Fig 3.1, as it can be seen from the waveform itself that it will be tough to get the features.

To remove the noise baseline wandering is done. In baseline wandering the fit of the wave is changed so that it becomes comparable with the default ECG waveform. After it is done using the curve fitting tool, it resembles pretty much to the default ECG waveform as shown in Fig 3.2.

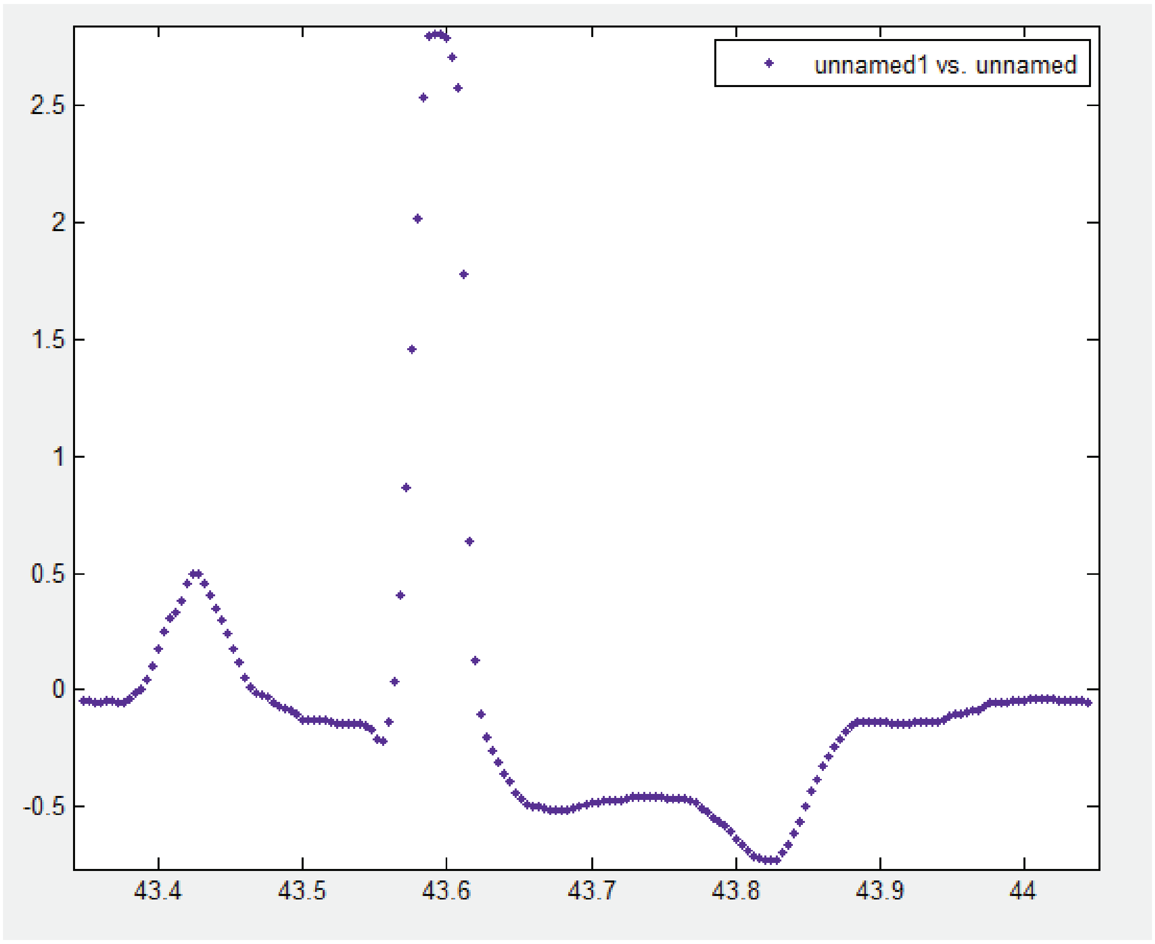

Figure 3.1: Before Baseline Wandering

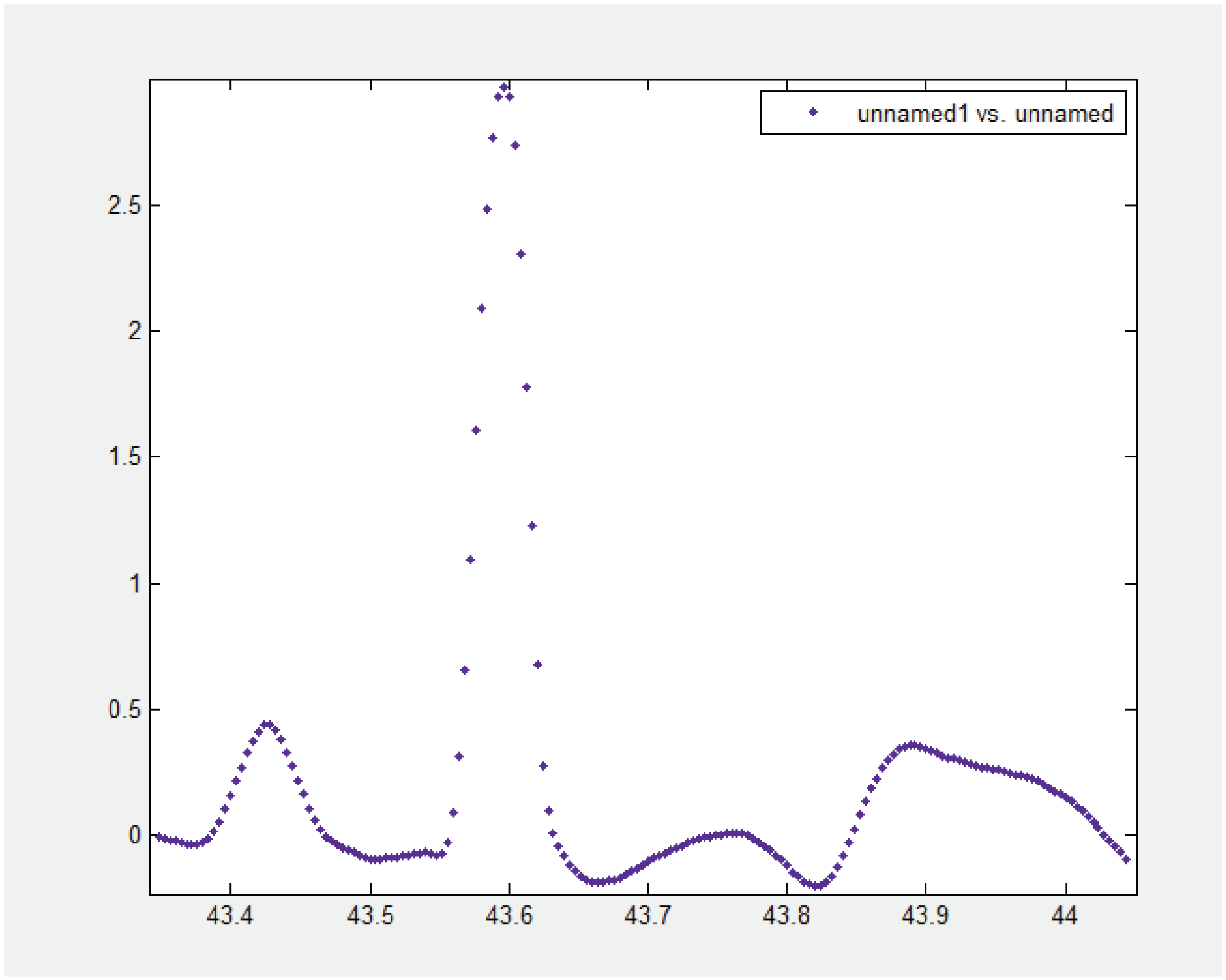

Figure 3.2: After Baseline Wandering

After the baseline wandering, the features are extracted accordingly. They are formed into a 28X8 matrix, on which PCA is later applied to reduce the dimensions and reduce it to the two main principal components that cover the maximum part of the data. The feature matrix extracted is shown below in Table 3.1.These features are then used for the classification of the waveforms.

Table 3.1: Feature Matrix After Extraction Of The Features

Sample No.	P (height)	P (width)	T (height)	T (width)	QRS (height)	QRS (width)	PR	QT
43_1	0.471	0.076	0.36	0.18	2.987	0.208	0.172	0.568
43_2	0.441	0.08	0.356	0.148	2.968	0.268	0.208	0.564
43_3	0.44	0.084	0.356	0.188	2.966	0.264	0.212	0.564
43_4	0.44	0.084	0.356	0.188	2.967	0.264	0.212	0.564
43_5	0.44	0.084	0.356	0.188	2.967	0.264	0.212	0.564
43_6	0.44	0.076	0.355	0.18	2.967	0.272	0.208	0.56
43_7	0.44	0.076	0.355	0.176	2.967	0.272	0.208	0.56
120_1	0.173	0.188	0.378	0.16	1.885	0.184	0.228	0.344
120_2	0.168	0.132	0.372	0.162	1.756	0.16	0.216	0.384
120_3	0.168	0.132	0.372	0.162	1.756	0.16	0.216	0.384
120_4	0.168	0.132	0.372	0.152	1.756	0.164	0.216	0.384
120_5	0.167	0.124	0.372	0.152	1.757	0.164	0.212	0.384
120_6	0.168	0.124	0.372	0.148	1.756	0.168	0.212	0.384
120_7	0.167	0.132	0.372	0.152	1.756	0.16	0.212	0.38
240_1	0.203	0.132	0.277	0.176	1.67	0.16	0.102	0.58
240_2	0.18	0.156	0.273	0.268	1.4	0.164	0.2	0.432
240_3	0.18	0.156	0.273	0.268	1.4	0.16	0.2	0.432
240_4	0.18	0.156	0.273	0.268	1.4	0.164	0.2	0.432
240_5	0.18	0.156	0.273	0.268	1.4	0.164	0.2	0.432
240_6	0.18	0.156	0.274	0.268	1.4	0.164	0.2	0.432
240_7	0.18	0.156	0.273	0.268	1.4	0.164	0.2	0.432
277_1	0.227	0.108	0.246	0.212	0.976	0.276	0.256	0.716
277_2	0.207	0.108	0.238	0.224	0.943	0.356	0.148	0.716
277_3	0.207	0.108	0.238	0.228	0.943	0.272	0.256	0.72
277_4	0.207	0.108	0.238	0.228	0.942	0.276	0.256	0.72
277_5	0.207	0.11	0.238	0.224	0.942	0.278	0.256	0.722
277_6	0.207	0.108	0.238	0.224	0.943	0.276	0.256	0.72
277_7	0.207	0.108	0.238	0.222	0.942	0.276	0.256	0.718

3.2 Application of PCA:

PCA is applied to get the principal components of the features extracted from the ECG waveforms. The principal components are shown below in Table 3.2.

Table 3.2: Principal Components of The Features Extracted

	Eigen Values (EV)	**Cumulative EV**	**% Variance**	**Cumulative Variance**
PC1	0.59323	0.59323	95.4335	95.4335
PC2	0.02443	0.61766	3.93046	99.364
PC3	0.00224	0.6199	0.36032	99.7243
PC4	0.00102	0.62092	0.1637	99.888
PC5	0.0005	0.62142	0.0799	99.9679
PC6	0.00013	0.62154	0.02023	99.9881
PC7	5.34E-05	0.62159	0.00859	99.9967
PC8	2.04E-05	0.62162	0.00328	100

As it can be properly seen from the table 3.2 itself that most part is covered by the first two principal components. Hence, the main aim of PCA is satisfied, as it reduced the data to a two dimension which when plotted will show the proper classification and clustering of the ECG. The syntax is shown below.

SYNTAX :

```
[COEFF,SCORE,latent] = princomp(X)
```

3.3 Description:

COEFF = princomp(X) performs principal components analysis on the n-by-p data matrix X, and returns the principal component coefficients, also known as loadings. Rows of X correspond to observations, columns to variables. COEFF is a p-by-p matrix, each column containing coefficients for one principal component. The columns are in order of decreasing component variance.princomp centers X by subtracting off column means, but does not rescale the columns of X. To perform principal components analysis with standardized variables, that is, based on correlations, use princomp(score(X)). To perform principal components analysis directly on a covariance or correlation matrix, use pcacov.

[COEFF,SCORE] = princomp(X) returns SCORE, the principal component scores; that is, the representation of X in the principal component space. Rows of SCORE correspond to observations, columns to components.

[COEFF,SCORE,latent] = princomp(X) returns latent, a vector containing the eigenvalues of the covariance matrix of X.

SCATTER PLOT:

After the desired Principal components are found, they are then plotted on a scatter plot which clearly classifies the waveforms. It gives us a proper idea as how many clusters are there, which could not have been done using the data as it was.

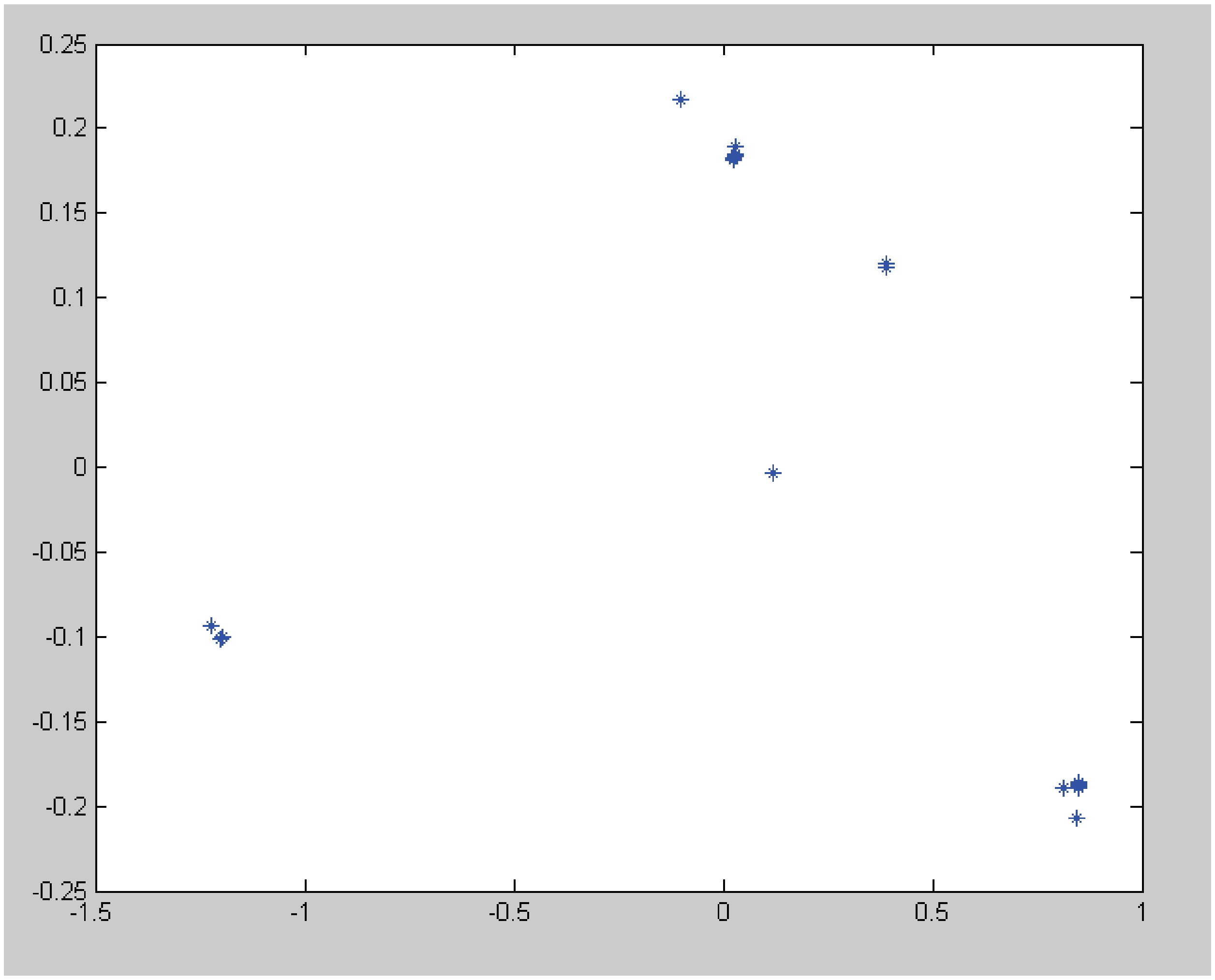

Figure 3.3: Scatter Plot of Scores of PC-1 Vs PC-2

3.4 PNN application:

PNN is applied to the data set extracted and it is trained and tested accordingly giving results fruitful for future work .The network can be used for various purposes with respect to the requirement.

Description:

Probabilistic neural networks (PNN) are a kind of radial basis network suitable for classification problems.

net = newpnn(P,T,spread) takes two or three arguments,

PR x Q matrix of Q input vectorsTS x Q matrix of Q target class vectorsspreadSpread of radial basis functions (default = 0.1) and returns a new probabilistic neural network. If spread is near zero, the network acts as a nearest neighbor classifier. As spread becomes larger, the designed network takes into account several nearby design vectors.

Examples:

Here a classification problem is defined with a set of inputs P and class indices Tc.

P = [1 2 3 4 5 6 7];

Tc = [1 2 3 2 2 3 1];

The class indices are converted to target vectors, and a PNN is designed and tested.

T = ind2vec(Tc)

net = newpnn(P,T);

Y = sim(net,P)

Yc = vec2ind(Y)

Algorithm:

newpnn creates a two-layer network. The first layer has radbas neurons, and calculates its weighted inputs with dist and its net input with netprod. The second layer has compet neurons, and calculates its weighted input with dotprod and its net inputs with netsum. Only the first layer has biases.

newpnn sets the first-layer weights to P', and the first-layer biases are all set to 0.8326/spread, resulting in radial basis functions that cross 0.5 at weighted inputs of +/- spread. The second-layer weights W2 are set to T.

CHAPTER-4

CONCLUSION

The entire Project gives a basic idea and a technique which can be used for varied purposes and mainly for disease diagnosis. Those who have a good amount of idea regarding the ECG waveforms and their characteristics due to diseases can use this technique to distinguish. The data taken is known and the classification is also known, but the methodology used can be applied for the real time classification. As ECG classification is a very important requirement for any kind of diagnosis, this method can be used and fruitful results can be generated. The algorithm given here can be used to classify the ECG data and get the desired results and outputs.

REFERENCES

1. Acharya, R., Bhat, P. S., Iyengar, S.S., Roo,A., & Dua, S. “Classification of heat rate data using artificial neural network and Fuzzy equivalence relation”, The Journal of the Pattern Recognition Society

2. Barro, S., Fernandez-Delgado, M., Villa-Sobrino, J.A., Regueiro, C.V., Sanchez, E., 1998. Classifying multichannel ECG patterns with an adaptive neural network. IEEE Eng. Med. Biol. Mag. 17 (1), 45–55.

3. Dickhaus, H., Heinrich, H., 1996. Classifying biosignals with wavelet networks. IEEE Eng. Med. Biol. Mag. 15 (5), 103– 111.

4. David C., Silverman. “Tutorial on Artificial Neural Networks”, SterlingGuidance on Corrosion and Materials Degradation.

5. Hu, Y.H., Pa lreddy, S., Tompkins, W., 1997. A patient adaptable ECG beat classifier using a mixture of experts approach. IEEE Trans. Biomed. Eng. 44 (9), 891–900.

6. Osowski, S., Linh, T.H., 2001. ECG beat recognition using fuzzy hybrid neural network. IEEE Trans. Biomed. Eng. 48 (11), 1265–1271

7. Ljmacphee., 2007. Herself’s Artificial Intelligence. Posted in Neural Networks - Topics in Artificial Intelligence.

8. Wasserman, P.D., 1993. Advanced Methods in Neural Computing, Van Nostrand Reinhold, New York.