Chapter1: INTRODUCTION AND LITERATURE REVIEW

1.1. Introduction

Now a days, in any power system process load forecasting has got a huge importance. Hence the question arises that what is the driving factor behind it? The answer is if we want to get the exact forecast for deciding the parameters of the power then load forecasting has to be motivated as the product as well as a trading article. All of us know that electricity is impossible to store, thus the estimate of electrical load in days to come has to be known in order to manage its production and consumption in an efficient way.

Generally we classify load forecasting in three methods. They are short, mid and long term models which is based on the span of time. If the prediction time span is of few seconds or minutes it can be said as very short term load forecasting (VSLTF) but if prediction time span is of hours to days it can be said as short term load forecasting. Here the demand is for the short term load forecasting, where we have already discussed that the needed prediction time changes between 5 minutes and 15 minutes.

Power system has been facing problems due to inefficient load forecasting in recent scenario. One reason is that recent technical innovative new ways have been brought up to solve the problem. The advancement in computer technology has expanded the potential for these methods working in a real time scenario. Another reason is that there is an international movement towards cut-throat competition in electrical power markets.

Although many forecasting techniques have been tested and proven successful, none has achieved a strong and authentic stature of a generally applied method. A reason is that the conditions and pre-requisite of a particular situation have significant effect on the choice of a desired model.

A significant amount of recently proposed approaches are based on Fuzzy logic (FL), Wavelet neural networks and Artificial Neural Networks (ANN) which have each yielded very hopeful results in solving the problem of Very Short term load forecasting (STLF).

Model combinations such as wavelet neural network output have yielded advances in reducing the forecasting error.

[1-2] presents comprehensive details on load forecasting and its need. [3] contains much needed information on the study of fuzzy logic. [4] reports different methods of load forecasting such as fuzzy and ANN which provide a good range of techniques to choose from. [5] has presented techniques for very short term load forecasting that is minute by minute. [6] has a very wide collection of study material and coding which is extremely helpful to understand a given topic. [7-8] give the methodology to be adopted for load forecasting and gives the right direction of approach to be implemented.

[9] contains data provided by TATA Power Delhi Distribution Limited on which the forecasting techniques are to implemented. [10-11] describes in detail about neural network, its working and applications. [12] presents the evaluation of the neural network as a load forecasting technique and a brief review about its application. [13] describes the use of back propagation algorithm for multilayer perceptron in neural networks. [14-18] present a classical and global approach to the application of neural network in short term and very short term load forecasting. [19] presents an example of a similar study done in the UAE.

[20] describes the wavelet and filter banks along with their relation and application. [21] proposes a methodology for the prediction of very short term load forecasting data using Wavelet Neural networks. [22-23] proposes the techniques of data pre-filtering for the prediction of 5 min data with minimum error. [24] provides the 5 minute load data of Bonneville Power Authority (BPA)

1.2. THE SCOPE OF WORK

This project studies the applicability of Fuzzy Logic, Artificial Neural Network and Wavelet Neural model on Load Forecasting.

With the help of this we can find the root work for the application of automatic forecasting, which can be applied in real-time environment.

A few properties are mentioned below which are taken as important:

- The model should be automatic and be able to adapt quickly to changes in the behaviour of load.
- The model is intended for use in many different cases. This means that generality is desired.
- Updating the forecast with new available data should be possible. The hours closest to the forecasting time should always be forecast as accurately as possible.
- This model should be reliable. Even exceptional circumstances must not give rise to unreasonable forecasts.
- Model is applicable to very short term load forecasting as temperature is neglected.
- The model should be easily attachable to an energy management system.

1.3. Organization of the project

The work done in this project consists of seven chapters which are given below.

Chapter 1 contains the literature review and the scope of the work.

Chapter 2 describes about the various aspects of load forecasting such as its importance, and the types and different methods available.

Chapter 3 describes fuzzy logic. It describes the inference systems of fuzzy logic, its membership functions and its applicability in load forecasting.

Chapter 4 describes artificial neural network, their training methods, multi-layer perceptron and its applicability in load forecasting.

Chapter 5 describes Wavelet Neural Network including the wavelet and wavelet families, the spike filtering techniques and a brief about filter bank.

Chapter 6 presents the case studies and results using the various load forecasting techniques.

Chapter 7 presents the conclusions.

CHAPTER 2 :Load Forecasting

2.1 INTRODUCTION

Load forecasting can be considered as a central and a vital process for planning and operation of electrical power utilities. Since last two decades, to tackle this problem, many techniques and their approaches have been investigated which differ in nature and applies different engineering considerations and economic analysis.

Load forecasting has constantly been significant for planning and operational decisions conducted by power utility companies. However, with deregulation of the industry, it has gained more importance. A ssupply and demand of electricity is fluctuating, the energy prices increases by a factor of ten or more during peak conditions. This makes it vitally important for the utilities application. It can help to estimate load flows thus easy to make a decision to prevent overloading. Proper implementation of such decisions leads to improvement of network reliability thus reduced occurrences of equipment failures or blackouts. It is also significant for contract evaluations or for evaluations of various sophisticated financial products based on energy pricing accessible by the market. In the freed economy, decisions on the capital expenditures based on long-term forecasting are also important than in a non-freed economy when rate increases that could be acceptable by capital expenditure projects.

2.2. FACTORS AFFECTING THE LOAD

Usually, load of an electrical power utility is poised of very different consumption units. A major part of the electricity is consumed by the industrial activities. Another part is used by private people in forms of heating, lighting, cooking, laundry, etc. Many services accessible by the society demands electricity, viz., street lightning, railway traffic lights etc.

Factors affecting the load depends basically on the particular consumption unit. The industrial load is typically determined by the level of the production. The load is regularly quite steady, and it is likely to guess its dependence on diverse production levels. However, from the point of view of the electricity utility selling, the industrial units typically add ambiguity in the forecasts. The problem is the possibility of unpredicted events, i.e., machine breakdowns, which can cause large random disturbances in load level.

As major part of consumption is because of private people and other small electricity customers, the typical approach in load forecasting is to concentrate on the collective load of the whole utility.

In short run, the meteorological situationsdo not causes large variations in thecombined load. In long run, the economic and demographic factors plays the most vital role in determining the progress of the electricity demand.

From the perspective of long term forecasting, the time factor is essential. By these, various seasonal effects and recurring behaviors (daily and weekly rhythms) as well as occurrences of lawful and religious holidays are meant.

The other factors causing disturbances can be called as random factors, which are usually small in the case of single consumer, although large social events and popular TV-programs add doubt in the forecasts. On the other hand, Industrial units can cause relatively large disturbances as compared to above.

2.3.TYPES OF LOAD FORECASTING

Depending upon the period of forecast, the load forecast is of three types :-Long term Load forecast, Medium term Load forecast and Short-term load forecast.

The Long-term load forecast takes pretty long time to plan, install an additional generating capacity. In this thesis work, short term load forecast is considered, which is essential for online control and security evaluation for a large system.

It takes fairly a long time to plan, install and commission an additional generating capacity. Conventionally, system expansion planning starts with a forecast of anticipated future load requirements. For optimal generation capacity expansion,proper long term forecasting is necessary.

One methodfor long term load forecasting is extrapolation, which is used by many utilities,. This involves fitting trend curves to basic historical data, adjusted to reflect the growth trend itself. Once the trend curve is known, the forecast is found by evaluating the trend curve function at the desired future point.

Another technique is Correlation which relates system leads to various demographic and economic factors. Typical factors like: population, employment, industrial licenses, appliance saturation, weather data etc. are used in correlation techniques. However, the forecasting the demographic and economic factors is rather difficult.

2.3.2MEDIUM TERM LOAD FORECAST

Medium term load forecast is similar to long term load forecasting. The only difference is of the time range. This takes the time period ranging from a couple of months to a year or so.

2.3.3SHORT TERM LOAD FORECAST(STLF)

A precise STLF is essentially for monitoring and controlling power system operations. The hourly load forecast with a lead-time up to one week in advance is necessarily for online solution of the scheduling problem. A 24-hour load forecast is needed for successful operation of power plants. One hour forecast is important for online time control and security evaluation of a large power system.

It generally involves physical decomposition of load into components. The load is decomposed into a daily pattern reflecting the difference in activity level during the day. A weekly pattern representing the day of the week effect on load is done. A trend component concerning the load pattern of previous loads of the week is observed. The random errors can be statistically analyzed to obtain a stochastic model for the error estimation.

When the forecasting is done for 5-15 minutes range then it is known as very short term load forecasting.

2.4. Load Forecasting Techniques

Load forecasting can be done through various techniques, viz., artificial intelligent techniques, numerical techniques and various numerical methods. Some of them are listed below

2.4.1Fuzzy Logic model

Fuzzy logic is a form of logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. In contrast with "crisp logic", where binary sets have binary logic, the fuzzy logic variables may have a membership value of not only 0 or 1 i.e. the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values of classic propositional logic. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions.

Fuzzy logic emerged as a consequence of the 1965 proposal of fuzzy set theory by LotfiZadeh. Though fuzzy logic has been applied to many fields, from control theory to artificial intelligence, it still remains controversial among mgoststatisticians, who prefer Bayesian logic, and some control engineers, whose prefer traditional two-valued logic.

2.4.2Artificial Neural Network model

A neural network is an interconnected assembly of simple processing elements known as neurons. The processing ability of the network is stored in inter - unit connection strengths called weights obtained by a process of learning from a set of training patterns."

It is a massively parallel distributed processor made up of neurons which has a natural propensity for storing experiential knowledge and making it available for use.

Neural model can be trained to forecast load by using different layers of neuron.

2.4.3Time Series models

In simplest form, a time series model takes them previous week's actual load pattern as a model to predict the present week's load. Alternatively, a set of load patterns is stored for typical weeks with different days load. These are then heuristically combined to create the forecast.

More commonly, a time series model is of the form:

$$z(t)=\sum_{i=1}^{N} a(i)\, f(t)+ v(t) \quad (2.1)$$

Where the load at time *t* is expressed as a weighted sum of explicit time functions, usually sinusoids with a period of 24 or 168. The coefficients *a(i)* are slowly varying constants being usually estimated through a linear regression or exponential smoothing. The modeling error v(t) is assumed to be white noise.

2.4.4Regression models

Regression models normally assume that the load can be divided into a standard load component and a component linearly dependent on some explanatory variables. The model can be written:

$$z(t)=b(t)+\sum_{i=1}^{N} a(i)y_i(t)+\varepsilon(t) \tag{2.2}$$

Where b (t) is the standard load, ε(t)is a white noise component, and y_i(t) are the independent explanatory variables. The most typeical explanatory varhiables are weather factors.

They model different consumer categories by separate regression models. The load is divided into a rhythm component and a temperature dependent component. The rhythm component corresponding to load for a certain hour in the averages temperature of the modeling period.

More complicated model variations have also been proposed. Some models use earlier load values as explanatory variables in addition to external variables.

Regression models are among the oldest methods suggested for load forecasting. They are quite insensitive to occasional disturbances in the measurements. The easy implementation is another of its strengths. The serial correlation, which is typical when regression models are used on time series, can cause problems.

2.4.5Stochastic models

The method appears to be a popular one that has been applied and is still applied to STLF in the electric power industry. The load series y(t), is modeled as the output form a linear filter that has a random series input, a(t) usually called white noise as showns in figure.

Fig2.1. Load Time Series Modelling

Depending on the characteristic of the linearized filter, different models are classified as Autoregressive (AR) process, Moving average (MA) process, Autoregressive Moving Average (ARMA) process. This is a very popular class of dynamic forecasting models.

The basic principle is that the load time series can first be transformed into a stationary time series (i.e. invariant t with respect to time) by a suitable differencing. Then the remaining stationary series can be filtered into white noise. The models assume that the properties of the time series remain unchanged for the period used in model estimation, and all disturbances are due to this white noise component contained in the identified process.

The stochastic time series models have many attractive features. First, the theory of the models is well known and therefore it is easy to understand how the forecast is composed. The properties of the model are easy to calculate; the estimate for the variance of the white noise component allows the confidence intervals for the forecasts to be created.

The model identification is also relatively easier. Established methods for diagnostic checks are available. Moreover, the estimation of the model parameters is quite straightforward, and the implementation is not difficult. The weakness in the stochastic models is in the adaptability. In reality, the load behavior can change quite quickly at certain parts of the year. While in **ARMA** models the forecast for a certain hour is in principle a function of all earlier load values, the model cannot adapt to the new conditions very quickly, even if model parameters are estimated recursively. A forgetting factor can be used to give more weight to the most recent behavior and thereby improve the adaptability.

Another problem is the handling of the anomalous load conditions. If the load behavior is abnormal on a certain day, this deviation from the no rmal conditions will be reflected in the forecasts into the future. A possible solution to the problem is to replace the abnormal load values in the load history by the corresponding forecast values.

2.4.6 State-space models

In the linear state-space model, the load at time *t* can be written:

$$Z\ (t)=C^T x\ (t) \qquad (2.3)$$

where,

$$X\ (t+1)=Ax\ (t)+Bu\ (t)+w\ (t) \qquad (2.4)$$

The state vector at time *t* is x (t), and u (t) is a weather variable based input vector. w(t) is a vector of random white noise inputs. Matrices **A**, **B**, and the vector **C** are assumed constants.

There exist a number of variations of the model. Some examples can be found.

In fact, the basic state-space model can be converted into an **ARMA** model and vice versa, so there is no fundamental difference between the properties of the two model types.

According to Gross and Galiana (1987), a potential advantage over **ARMA** models is the possibility to use *prior* information in parameter estimation via Bayesian techniques. Yet, they point out that the advantages are not very clear and more experimental comparisons are needed.

2.4.7 Wavelet Neural Network

Wavelet neural network is the approach which first modifies the input and then uses neural network. It is generally used when the input has spikes, i.e.one of the input vary drastically in comparison to the neighbor input. Hence a pre-filtering of the inputs is done through wavelet using different functions of the wavelet family and then neural network is applied for satisfactory predictions.

Chapter 3: Fuzzy Logic

3.1 INTRODUCTION

We use fuzzy logic basically to solve problems which are approximate instead of accurate. It is a type of logic which is evolved from fuzzy set theory. Unlike the binary logic which have binary sets having values 0 and 1, generally said as "crisp logic", fuzzy logic variables can lie in a range from 0 to 1, but are not limited to the two truth values as in classical logic. Moreover, specific functions are used to manage the extent of accuracy when semantic variables are used.

A proposal given by Lotfi Zadeh in the year 1965 about fuzzy set theory gave rise to the concept of fuzzy logic. Apart from the fact that fuzzy logic has been used in various spheres such as artificial intelligence, control theory, etc., still it is contentious among a lot of statisticians. They favor Bayesian logic. Similarly, a few control engineers favor classical two-valued logic instead of fuzzy logic.

3.2 FUZZY INFERENCE SYSTEM

Fuzzy inference is a method of formulating the mapping from a given input system to an output with the help of fuzzy logic. Hence with the help of the mapping , a foundation is obtained with which we can make decisions and patterns can be recognized.

The method of fuzzy inference includes all of the parts which are explained in the former sections which are 1) Membership, 2) Logical Operations, and 3) If-Then Rules. Application of fuzzy inference system can be done in two types in the toolbox: 1) Mamdani-type and 2) Sugeno-type. Based on the type of outputs which are obtained the mentioned types of inference system variegate.

3.2.1. FUZZIFICATION

It's the process of converting conventional values into grades of membership for linguistic terms of fuzzy sets. It decomposes input and output system into one or more fuzzy logic sets.

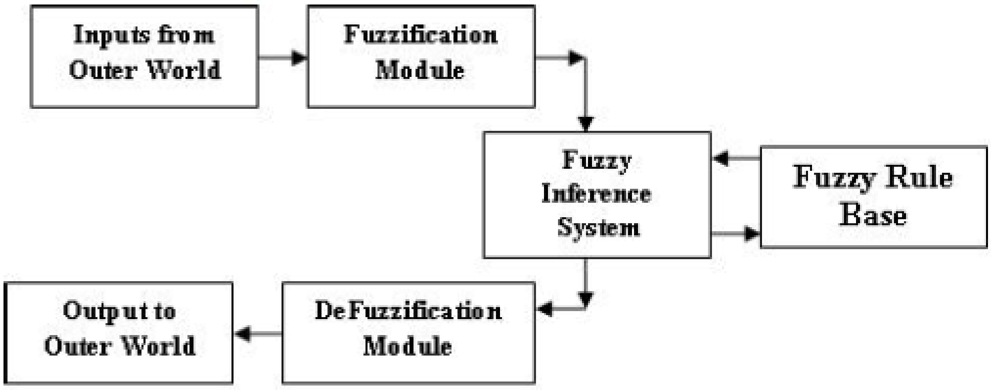

Fig 3.1: Flow and processing data in FIS

3.2.2. MEMBERSHIP FUNCTIONS

The membership function represents the participation of each magnitude in the form of graphical view which associates the of each inputs, that are processed, defines functional overlap between the inputs and determines the response across output. Once the functions are inferred, scaled, and combined across the output , they are de-fuzzified into a crisp output which drives the system. There are different form other memberships functions as they are associated with each input and output response.

Some features related to fuzzification are:

- **SHAPE** - Triangular is most common shape, but bell, trapezoidal, sine and exponential shapes are used. More complex functions can be used for fuzzification but it requires greater computing to implement it.
- **HEIGHT** or magnitude (usually normalized to 1)
- **WIDTH** (of the base of function)
- **SHOULDERING** (locks height at maximum if an outer function. Shouldered functions evaluate as 1.0 past their center)
- **CENTER** points (center of the member function shape)
- **OVERLAP** (N&Z, Z&P, typically about 50% of width but can be less).

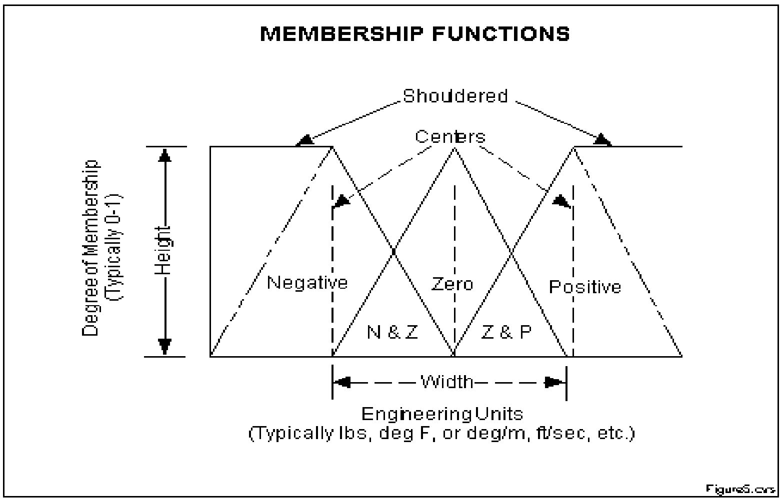

Fig 3.2: Membership functions

Fig 3.2 above, illustrates the features of the triangular membership function which is used in this example because of its simplicity to solve mathematically. Other shapes can be used but the triangular shape lends itself to this illustration. By choosing the selected input parameter from error or error-dot via the horizontal axis and projecting it vertically to the upper boundary of the membership functions the degree of membership (DOM) may be determined.

Different types of membership functions are discussed below:

The most commonly used in practice are triangles, trapezoids, bell curves, Gaussian and sigmoidal function [8]

1. **Triangular membership function**

A triangular membership function is specified through three parameters as a,b,c which can be seen in fig3.3.

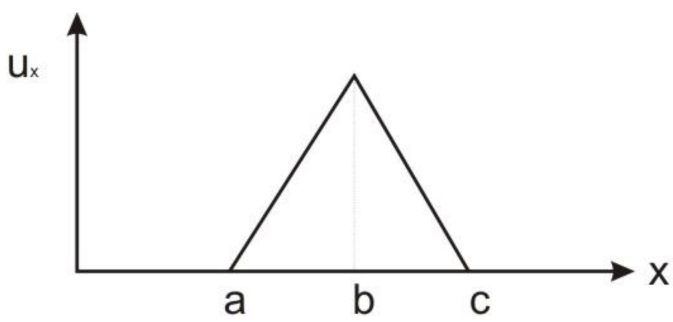

Fig 3.3: Triangular Membership function

Mathematically it can be defined as

$(x{:}a,b,c) = 0 \qquad x < a$

$(x-a)/(x-b) \qquad a \le x \le b$

$(c-x)/(c-b) \qquad b \le x \le c$

2. Trapezoidal membership function

A trapezoidal function is specified by four parameters in form of a,b,c,d which is shown in fig.3.4

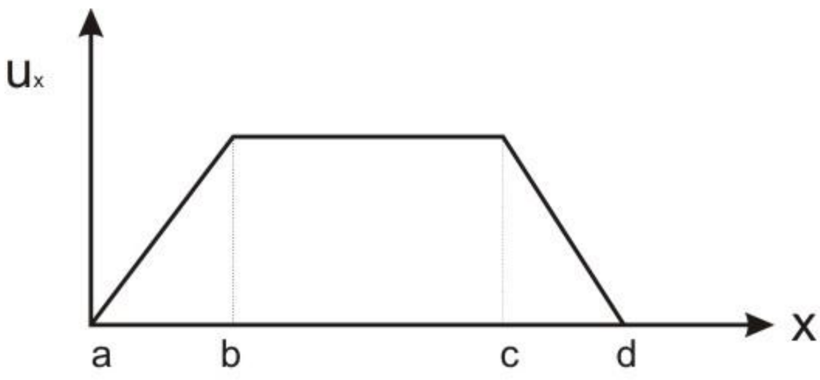

Fig 3.4: Trapezoidal Membership function

Mathematically defines as:

$(x{:}a,b,c,d) = \quad 0 \qquad x<a$

$(x-a)/(x-b) \qquad a\le x\le b$

$1 \qquad b\le x\le c$

$(d-x)/(d-c) \qquad c\le x\le d$

3. Gaussian Membership Functions

A Gaussian membership function is specified by only two parameter as c, k which is shown in fig.3.5

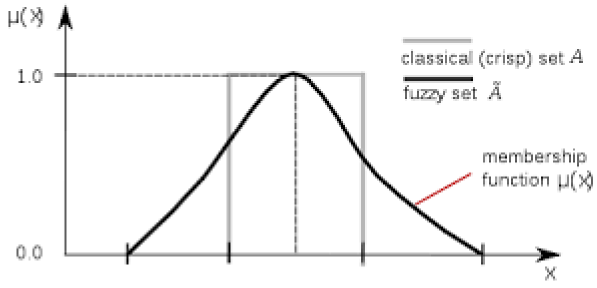

Fig 3.5: Gaussian membership function

Mathematically defined as-

Gaussian (x: c, k)= exp(- $\frac{(x-c)^2}{k^2}$)

where c and k denote the center and width of function respectively.

4. Sigmoidal Membership function

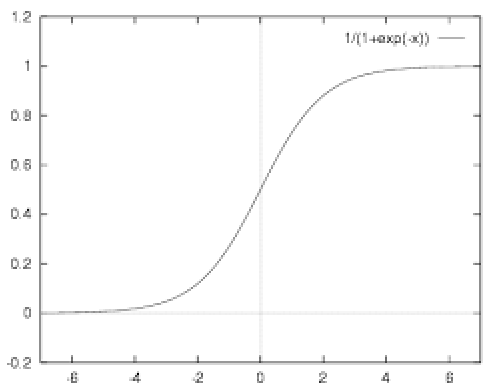

Fig 3.6: Sigmoidal membership function

A sigmoid function mathematically expressed as:

Sigm(x: a, c) = $\frac{1}{1+e^{-a(x-c)}}$

3.2.3. RULE BASE

The rules related to fuzzy set theory is that it defines operators of fuzzy on fuzzy sets, but the problem associated with this rule is that the appropriate fuzzy operator may not be known through this. Due this reason, fuzzy logic usually uses IF-THEN rules, or constructive models which are equivalent to it, like fuzzy associative matrices. Rules are usually enunciated in form of : IF *variable*, IS *property* & THEN *action.*

3.2.4. INFERENCING

For doing inferencing in fuzzy logic, MAX-MIN METHOD is used for testing the magnitude of each rule and select the highest value from it. The "fuzzy centroid" is the area which is under taken for the output in horizontal coordinates. It exactly does not combine the effects of all rules which are applicable to it, but the imp function which is done by it is that it's continuous in nature and can be easily implemented.

In fuzzy logic to fit the member of a function to its respective peak value, MAX –DOT or MAX-PRODUCT method is used and "fuzzy centroid" of the horizontal coordinates is taken as area under the function for getting its output.

3.2.5. DEFUZZICATION

There are two common defuzzification techniques which are as follows:

1) Mean of maximum (MOM)

This method, calculates the average of output values who have highest possibilities of degrees. Suppose "y is A" is a fuzzy connection which is to be defuzzified. So, it may be mathematicalluy derived as the MOM defuzzification method

$$\text{MOM (A)} = \sum_{Y*\in p} y*/\text{p} \qquad \text{..........3.1}$$

where

p is the set of output values y with the possibilities degree in A

$$\text{P}= \{\text{y*}!\mu_a(\text{y*})=\text{sup}\mu_a(\text{y})\} \qquad \text{..….3.2}$$

Where P is an interval, the result of MOM defuzzification is the midpoint in the interval of he system.

2) Centre of area (COA)

The Centre of area (COA) also called as centre of gravity ,or centroid method in literature. It’s one of the most popular defuzzification technique. This method, takes the entire possibility distribution value in calculating its representative points. Fuzzy logic controller computes the area under the membership function which is also in the range of output variable.

3.3. Methodology

Every 15 minute load data is taken for Monday to Friday. The loads on weekdays are same i.e the load for Monday, Tuesday, Wednesday and Thursday are comparatively close rather than on Saturday and Sunday.

Hence we take the four input from Monday to Thursday and we predict the output that is the load for Friday. The loads are first normalised according to the formula

$$L_s =((Y_{max}-Y_{min})/(L_{max}-L_{min}))(L-L_{min})+Y_{min} \quad (3.3)$$

Y_{max} = maximum Linearized load limit

Y_{min} = minimum Linearized load limit

L_{max} = maximum load limit

L_{min} = minimum load limit

After linearization of load membership functions are defined which are chosen to be triangular and Gaussian. Rules are defined after the formation of membership function.

3.4. SYSTEM UNDER CONSIDERATION

- A good quality of historical data for input parameters for the last few years of Pitampura, INDIA (from 2008-2012) were collected.[11]
- History data for electrical load of TPDDL (of April 2012) were also collected.
- We consider four inputs for load forecasting
- By analyzing the historical data’s the inputs are classified as follows:

○ Monday Load - 17 membership functions

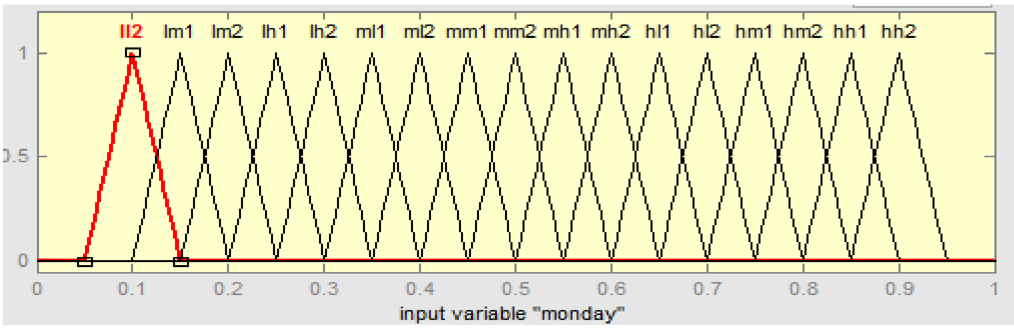

Fig 3.7: Membership Function for Load on Monday

○ Tuesday Load - 17 membership functions

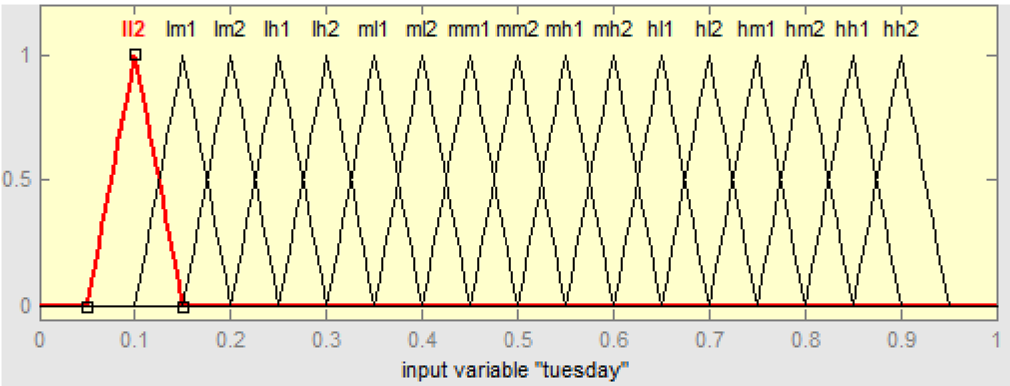

Fig 3.8: Membership Function for Load on Tuesday

○ Wednesday Load - 17 membership functions

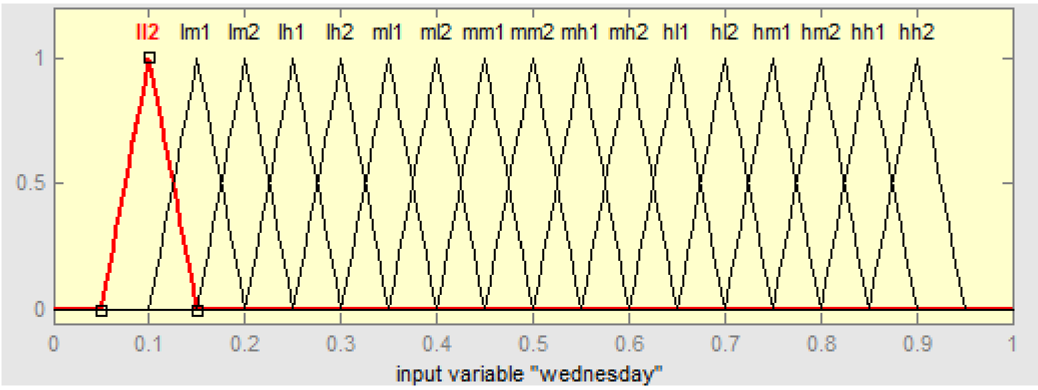

Fig 3.9: Membership Function for Load on Wednesday

- Thursday Load – 17 membership functions

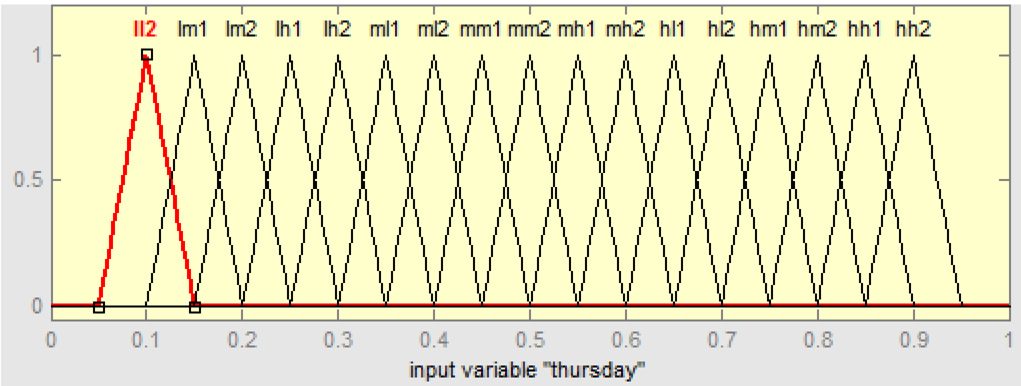

Fig 3.10: Membership Function for Load on Thursday

- For more accuracy, the output loads are classified into 17 membership functions.

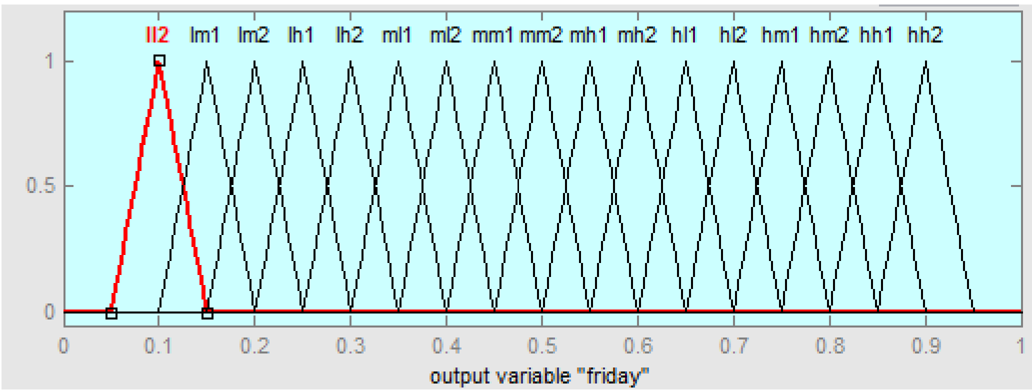

Fig 3.11: Membership Function for Load on Friday

- The data available is of fifteen minutes interval from 11 April 2012 to 15 April 2012 of PITAMPURA, DELHI

- The data available is given in Table I and the normalized data is given in Table II.

Chapter 4: Artificial Neural Networks

4.1. INTRODUCTION

"A neural network is an inter-linked assemblage of simple processing elements known as neurons. The processing capability of the network is confined in inter - network connection strengths known to be weights obtained by a method of learning from a set of training patterns."

It is a massively parallel distributed processor composed of neurons which has a natural inclination for storing experiential data and making it accessible for use.

It resembles the brain in two respects

- Information is obtained by the network from its atmosphere through a learning process.
- Interneuron connection strengths also known as the synaptic weights are used to obtain the acquired information.

The Artificial Neural Networks are stimulated from the human brain which consists of structural constituents called neurons (nodes, units, processing elements etc).

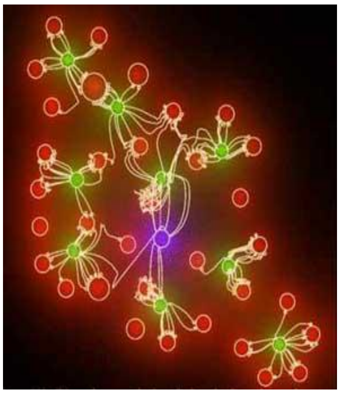

Fig 4.1: Artificial Neural Network

Neural networks can be divided into two as a single layered & multi layered model.

- Single layered models: These networks consist of only single layer of neurons, to which the input is specified and an output is obtained.

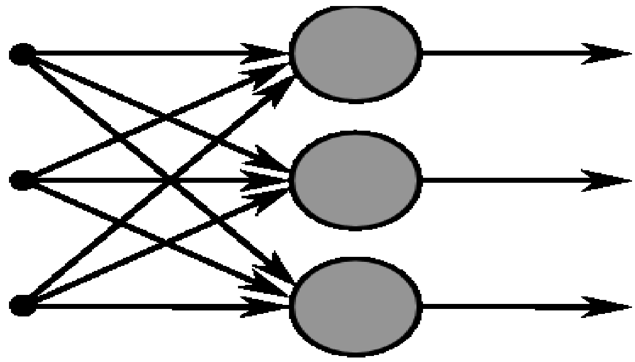

Fig 4.2: Single layer multi-network model

These networks can have three or more than three layers. For models such as these, there are basically three layers input layer, hidden layer and the output layer.

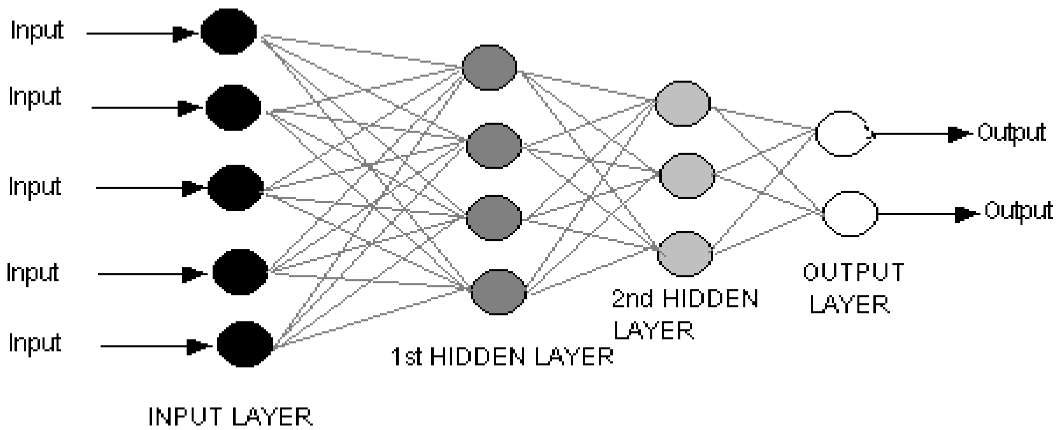

Fig 4.3: Multi layer multi-network model

4.2. TRAINING METHODS

All neural network models after modelling must undergo training, and there are three distinct classifications through which networks may be trained.

- **Supervised training method**: In this method, the network is made to learn by providing an external teacher. Here a set of example pairs are given (x,y), x belongs to X and y belongs to Y, and the aim is to find a function f : X->Y in the allowed class of functions that matches the examples. Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation).

- **Unsupervised training method**: In this method we are given some data *x*, and the cost function to be minimized can be any function of the data *x* and the network's output, *f*. Here an external teacher is not used. Tasks that fall within the paradigm of unsupervised learning are in general estimation problems.
- **Reinforcement learning**: It projects the learning aspects based on the network's actions and are represented either as good action or bad action. Tasks that fall within the model of strengthening the learning are control problems and sequential decision making responsibilities.

4.3 FEED FORWARD PERCEPTRON MODEL

The first original percepetron model was modeled by Frank Roseanblat in 1958. This model is composed of three layers, (1) a 'retina' that disperses the inputs to the next layer, (2) 'association units' that merge the inputs and weights and trigger the threshold activation function feeding to the output layer, (3) the output layer which merge the values.

Multi-Layer Percepetron network is the most accepted neural network type and many of the reported neural network load forecasting models are planned on it. The basic unit (neuron) of the network is a *percepetron*. This is a computation unit, which produces its output by processing a linear combination of the input signals and by transforming this by a function called *activity function*. The output results of the percepetron as a function of the input signals can be written as:

$$y = \sigma\left(\sum_i^n w_i x_i - \theta\right), \tag{4.1}$$

where

y = output signals

x_i = input signals

w_i = neuron weights

Θ = bias term (another neuron weight)

Activation function: A function used to transform the activation level of a unit (neuron) into an output signal. Typically, activation functions have a "squashing" effect. There are a wide range of activation functions. Some of the most important and widely used functions are threshold function, sigmoid function, piecewise linear function and hard limiter function. Possible forms of the activity function are linear function, step function, logistic function and hyperbolic tangent function.

The MLP network consists of multiple layers of neurons. Each neuron present in a certain layer is connected to each neuron of the next layer. There are no feedback connected layers. A three-layer MLP network is shown in figure 4.4

The most often used MLP-network consists of three layers: an input layer, one hidden layer, and an output layer. The activation function used in the hidden layer is usually nonlinear (sigmoid or hyperbolic tangent) and the activation function in the output layer can be either nonlinear (a nonlinear-nonlinear network) or linear (a nonlinear network).

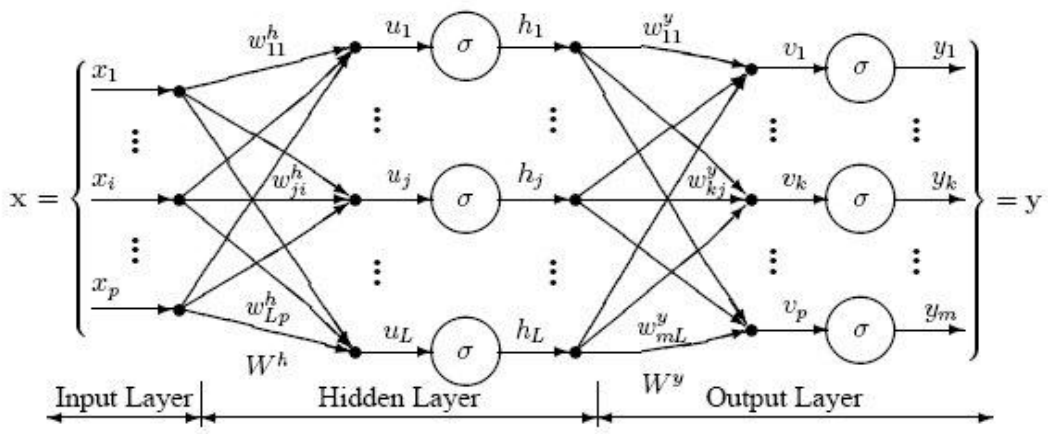

Fig 4.5: A Perceptron network with three layers

Input Layer: A vector of predictor variable values ($x_1 \ldots x_p$) is presented to the input layer. The input layer (or processing before the input layer) standardizes these values so that the range of each variable is -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the *bias* that is fed to each of the hidden layers; the bias is multiplied by a weight and added to the sum going into the neuron.

Hidden Layer: Arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight (w_{ji}), and the resulting weighted values are added together producing a combined value u_j. The weighted sum (u_j) is fed into a transfer function, σ, which outputs a value h_j. The outputs from the hidden layer are distributed to the output layer.

Output Layer: Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight (w_{kj}), and the resulting weighted values are added together producing a combined value v_j. The weighted sum (v_j) is fed into a transfer function, σ, which outputs a value y_k. The y values are the outputs of the network.

4.4 Techniques Used

Multiple layer perceptron's have been applied successfully to solve some difficult diverse problems by training them in a supervised manner with a highly popular algorithm known as the error back-propagation algorithm. This algorithm is based on the error-correction learning rule.

Error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an input vector is applied to the nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the weights of the networks are all fixed. During the backward pass, the weights are all adjusted in accordance with an error correction rule. The actual response of the network is subtracted from a desired response to produce an error signal. This error signal is then propagated backward through the network, against the direction of synaptic connections. The weights are adjusted to make the actual response of the network move closer to the desired response. A multilayer perceptron has three distinctive characteristics:

1. The model of each neuron in the network includes a nonlinear activation function. The sigmoid function is commonly used which is defined by the logistic function:

$$Y=\frac{1}{1+\exp(-x)} \tag{4.2}$$

Another commonly used function is hyperbolic tangent :

$$y = \frac{1-\exp(-x)}{1+\exp(-x)} \tag{4.3}$$

2. The network contains one or more layers of hidden neurons that are not part of the input or output of the network. These hidden neurons enable the network to learn complex tasks.

3. The network exhibits a high degree of connectivity. A change in the connectivity of the network requires a change in the population of their weights.

4.4.1 Back propagation

To demonstrate the method a three layer neural network is depicted in the figure below which is having two inputs and one output .

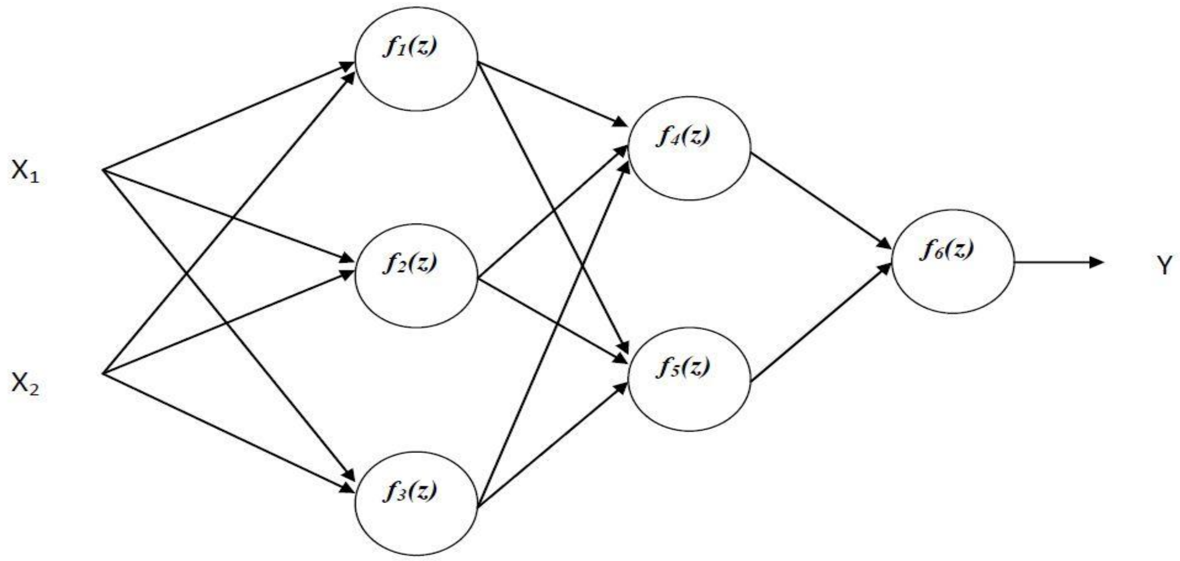

Fig 4.6: Three layer neural network with two inputs and single output

Signal z is taken as the adder output signal, while for representing non linear element $y = f(z)$ is used. We can also take Signal y as the output signal of neuron. $x1$ and $x2$ are taken as the input signals in the training data set. These are imputed with desired output y' or can be said as the corresponding target. The network training is considered to be an iterative process. From the given training data set, new data is used for changing the weights coefficients of nodes for every iteration. The meaning of the symbols *wmn* is weights of connections between output of neuron *m* and input of neuron *n* in the next layer. Symbols *yn* stands for the output signal of neuron *n*.

$$Y_1 = f_1(w_{11}x_1 + w_{21}\ x_1)\ \text{..........................(i)}$$

$$Y_2 = f_2(w_{12}\ x_1 + w_{22}\ x_2)\ \text{.........................(ii)}$$

$$Y_3 = f_3(w_{13}\ x_1 + w_{23}\ x_2)\ \text{..........................(iii)}$$

$$Y_4 = f_4(w_{14}\ y_1 + w_{24}\ y_2 + w_{34}\ y_3)\ \text{..........(iv)}$$

$$Y_5 = f_5(w_{15}\ y_1 + w_{25}\ y_2 + w_{35}\ y_3)\ \text{............(v)}$$

$$Y_6 = f_6(w_{16}\ y_4 + w_{56}\ y_5)\ \text{.........................(vi)}$$

The desired output value (the target), which is found in training data set. The difference is called error signal *δ* of output layer neuron.

$$\delta = y' - y$$

$$\delta 4 = w_{46}\ \delta$$

$$\delta 5 = w_{56}\ \delta$$

$$\delta 3 = w_{34}\ \delta_4 + w_{35}\ \delta_5$$

$$\delta 2 = w_{24}\ \delta_4 + w_{25}\ \delta_5$$

$$\delta 1 = w_{14}\ \delta_4 + w_{15}\ \delta_5$$

We can change the weights coefficients of each neuron for input node, when we calculate the error signal for every neuron. In the given mathematical equation *df(z)/dz* is derivative of neuron activation function.

Delta rule defines the connection of neurons between two weights ie., *i* and *j* of weight correction ($w_{ij}(n)$), where Weight correction is defined as the product of three, these are the learning rate parameter, local gradient and input signal of neuron i

$$\Delta w_{ij}(n) = \eta\ .\ \delta_i(n)\ .\ y_j(n)$$

Neuron *i* can be either an output node or hidden node, this decides the local gradient $\delta i(n)$:

1. If neuron *i* is an output node, *δi(n)* equals the product of the derivative *dfi(z)/dz* and the error signal *ei(n)*, both of which are associated with neuron i.

2. If neuron *j* is a hidden node, *δi(n)* equals the product of the associated derivative *dfi(z)/dz* and the weighted sum of the *δs* computed for the neurons in the next hidden or output layer that are connected to neuron *j*.

4.4.2 Levenberg-Marquardt Back-Propagation Algorithm

Levenberg – Marquardt algorithm is specifically designed to minimize sum-of-square error functions.

$$E = \tfrac{1}{2} \sum k (e_k)^2 = \tfrac{1}{2} \|e\|^2 \tag{4.4}$$

Where e_k is the error in the k_{th} exemplar or pattern and e is a vector with element e_k. Updating of the weights therefore involves the inverse Hessian or an approximation thereof formsnon-linear network. The Hessian is relatively easy to compute, since it is based on first order derivatives with respect to the network weights that are easily accommodated by back propagation. Although the updating formula could be applied iteratively to minimize the error function, this may result in a large step size, which would invalidated the linear approximation on which the formula is based.

In the Levenberg-Marquardt algorithm, the error function is minimized, while the step size is kept small in order to ensure the validity of the linear approximation. This is accomplished by use of a modified error function of the form.

$$E = \tfrac{1}{2}\left\| e(j) + \partial e_k / \partial w_i \left(w(j+1) - w(j) \right) \right\|^2 + \lambda \left\| w(j+1) - w(j) \right\|^2 \tag{4.5}$$

Minimizing the modified error with respect to w(j+1) gives

$$w(j+1) = w(j) - \left(Z^T Z + \lambda I \right)^{-1} Z^T e(j) \tag{4.6}$$

very large values of l amount to standard gradient descent, while very small values l of amount to the Newton method.

4.6. Methodology

- **a)** Import Data:-The data set used is in the Table 5 which shows historical quarter- hourly load observations from the TPDDL Pitampura for the year 2012, April.
- **b)** Preprocessing Data :- The dataset is divided into 2 sets, a training set which includes 70% of the sample and 15% for validation and 15% for test.
- **c)** Initialize Network: - In the case of MLP we initialize a default network of two layers with difference between input and output layer neurons.
- **d)** Train Network: - We use the “mean absolute error” performance metric for training of the network.
- **e)** Forecast: - Once the model is built, perform a forecast on the independent test set.

Chapter 5: Wavelet Neural Network

5.1. Introduction

Wavelets are a category of function which is implemented to localise the given function in both the position and scale. Signal processing, time series analyses are some of its typical applications. Wavelets defines the basic concept of the wavelet transform (WT) which shortens up data of functions or operators into variable frequency components, and then studies each componont with a resolution which is coordinated to its position and scale. In the context of signal processing, the wavelet transform is dependent upon two variables

- scale (or frequency)
- time

There are generally two types of wavelet transforms:

- continuous wavelet transform
- discrete wavelet

The first is defined for working with functions that has span of the whole real axis. Another wavelet transform deals with functions that are designed over a range of integers (generally t=1,2,3…..,N-1, where N denotes the number of values in time series).

5.2. Wavelet

A wavelet is a 'small wave' function, usually represented as $\psi(u)$. A small wave growth and decay has a finite span of time, as compared to a 'large wave', for example a sine wave, whose growth and decay is repeated over a period of time till infinity.
For a function $\psi(u)$ which is defined over the real axis (that is from $-\infty$ to $+\infty$) can be classified as a wavelet function if it satisfies the following three conditions:

(1) The integral of $\psi(u)$ is zero

$$\int_{-\infty}^{\infty} \psi(u)du = 0$$

(5.1)

(2) The integral of the square of $\psi(u)$ is unity:

$$\int_{-\infty}^{\infty} \psi^2(u)du = 1 \tag{5.2}$$

(3) Admissibility Condition:

$$C_\psi \equiv \int_0^{\infty} \frac{|\Psi(f)|^2}{f} df \tag{5.3}$$

Equation (5.1) shows that any value of the wavelet function ψ above zero, must be cancelled out by the values below zero. Clearly the line $\psi(u) = 0$ will satisfy this, but equation (5.2) tells us that ψ must make some finite values away from zero. If the permissibility condition can also be fulfilled then the signal to be analysed can be easily reconstructed from its continuous wavelet transform. One of the basic wavelet functions is the Haar wavelet Fig 5.1 named after A. Haar who developed it in 1910. A Haar wavelet can be described as:

$$\psi^{(H)}(u) \equiv \begin{cases} +1 & \text{if } 0 \leq u < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \leq u < 1 \\ 0 & \text{else} \end{cases}$$

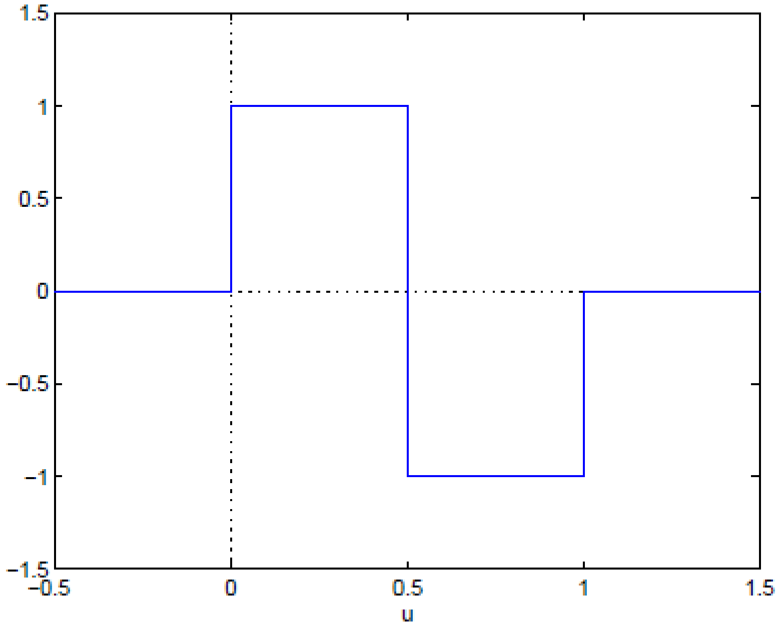

Fig 5.1: HAAR wavelet

5.3. Wavelet neural network

A wavelet neural network (WNN) mingles the theory of wavelets as well as neural networks. A WNN is defined to have of a feed-forward neural network model, which also has a few hidden layer, and An orthogonal normal wavelet family's activation function are incorporated into it. Among various applications of the wavelet neural network "Function estimation" is one. A wavelet network is guided to memorize all the basic elements of those functions whose value is given through a number of experiments , thus this helps to find the exact value for any given input.

The model of a wavelet neural network is a lot similar to that of a one and a half layer neural network. A neural network which is forward feeding, can use either one or even more than oneas its inputs, which must have one of the layers as hidden layer and thus the output layer of these network must have either one or more linear combinations or summations. Neurons are basically incorporated in the hidden layer. Wavelet basis is used for drawing the activation functions of these neurons and hence are termed as wavelons.

There are two main techniques for modelling wavelet neural networks.

• Firstly the wavelet and neural network processing are done separately. The input signal is first decomposed using some wavelet family function by the neurons in the hidden layer. The wavelet coefficients are then output to one or more summers whose input weights are modified in accordance with some learning algorithm.

• The second type combines the two theories. In this case the transformation and dilation of the wavelets along with the summing weights are modified in accordance with some learning algorithm. In general, when the first approach is used, only dyaadic dilations and transformation of the first wavelet form the wavelet basis. This type of wavelet neural network is usually referred to as a wavenet.

5.4. Wavelet Family

The selection of wavelet is constrained by the signal attributes and the nature of the application. If we understand the property, analysis and synthesis of a wavelet, we can select a wavelet that is most advantageous for our application.

Wavelet families vary in terms of several significant characteristic . Examples include:

- Support of the wavelet in time and frequency and rate of decay.
- Symmetry or anti-symmatry of the wavelet. The associated perfect reconstructiion filters have linear phase.
- Number of desertion moment. Wavelets with increasing numbers of desertion moments result in sparse representations for a large class of signals and images.
- Regularity of the wavelet. Smoother wavelets provide sharper frequency resolution. Additionally, iterative algorithms for wavelet construction converge faster.

- Subsistence of φ, which is a scaling factor.

5.5. Spike Filtering

The wavelet neural network can be used to forecast load however it is generally used where the spike is observed in the input. The spike can be said as micro spike if the span is 4 seconds whereas on the span of 5 minutes it is called macro filtering.

The way is to evaluate known and predicted loads, and if the absoluete values of the differences are greater than a threshold, spikes are confirmed and then replaced by predicted values. Another way is to replace pragmatic spikes by zeros which is then fixed by using a splining algorithm. These methods are important. However, they are prone to errors due to the indecisive characteristic of the load data and the various magnitudes and widths of spikes. Spikes replaced by arbitrary values may degrade future predictions. Therefore, spikes have to be further analyzed, and effectual ways are highly needed for filtering them out. Spike filtering has also been reported for short-term load forecasting. In comparison to VSTLF, spikes in STLF have different features with respect to magnitudes and widths because of the integrative nature of short-term load data and the fact that most spikes should have been removed before STLF is performed. The simple techniques consisting of if-then rules, low pass filtering and NN based self-filtering were used to handle STLF spikes. The key idea for filtering out spikes is to detect a pair of edges, and fix the loads between the two edges with linear interpolation values. This method is only applied to the loads at the 5-min resolution because macro spikes at the lower resolution may become micro spikes after integration. To detect edges, the first-order differencing transformation is applied to the load series at the 5-min resolution. The edge is said to be observed when the absolute value of the difference surpasses the threshold.

5.6. Filter Bank

To implement the two-level wavelet transform, a three-channel filter bank is used as shown in Fig 5.2. The high frequency channel consists of the analysis and synthesis stages. At analysis stage, a high pass filter (a wavelet function that plays a role in anti-aliasing) G_1 filters out the low frequency component. A down-sampling step then removes the odd-numbered data points. At the synthesis stage, the up-sampling step pads zeros to down-sampled data to recover the data length. A high pass filter H_1 then removes the replicas of signal spectrum caused by up-sampling. Similarly, the low-high frequency channel uses a low pass filter G_0 to compute the general trend, and then holds the even-numbered points. Next, these points are further decomposed into two parts. The low-high part convolves with G_1 and then takes steps similar to those for the high frequency channel. To recover the initial input length, the output from H_1 has to be up-sampled and convolve with H_0. These are the steps to produce the LH frequency component. The same is true for the low-low frequency channel. Filter $G_{0,}$,G_1, H_0 and H_1 have to satisfy perfect reconstruction and orthogonality.

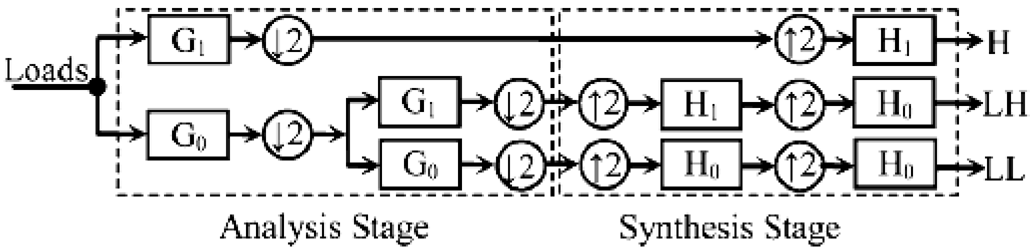

Fig 5.2: Three Channel Filter

To reduce the distortion, it is necessary to extend the signal beyond the boundaries. In the high frequency channel shown in Fig. 5, the distortion length for a convolution between the input loads and G_1 is (lw-1) based on the convolution theory, where lw is the filter length. Down-sampling and up-sampling do not produce the distortion. H_1 introduces another distortion with the same length. The total distortion length is 2(lw-1). The low-high frequency channel sequentially convolves the inputs with four filters ($G_{0,}$,G_1, H_0 and H_1) with a final length 4(lw-1) of the distortion , doubling that of the high frequency channel. The same is true for the low-low frequency channel. The distortion length is thus roughly doubled for a component which is further decomposed one more level. To make sure that at least one value is not affected by distortion, the load inputs to NN need to be padded. The padding length has to be equal to or greater than the distortion length.

$$lx = (lw\text{-}1)\ .\ 2^{\text{lvl}}$$

lx = distortion length which is also indicating minimum padding length

lvl = level of decomposition

The Db members which are analysed are from Db2 – Db20 (even index are taken only). The index no. also shows the length *lw*. The Db member cannot be taken large else minimum padding length will increase. Once *lvl* is fixed *lw* can be varied according to wavelet index. Hence the error can be calculated with different wavelet family and indices. The most suitable result can be obtained henceforth.

5.7. Methodology

a) Import Data:-The data set used is taken from Bonneville balancing authority, USA dated 16 February 2015 up to 20 February 2015

b) Wavelet application:- Wavelet is applied to remove the irregularities among the load with a specific wavelet family.

c) Preprocessing Data:- The dataset is divided into 2 sets, a training set which includes 70% of the sample and 15% for validation and 15% for test.

d) Initialize Network: - In the case of MLP we initialize a default network of two layers with difference between input and output layer neurons.

e) Train Network: - We use the "mean absolute error" performance metric for training of the network.

f) Forecast Load:- the load is forecasted if not satisfied the wavelet family or function is changed and steps are repeated from step (c).

Chapter 6: Case Studies and Results

CASE I: Load forecasting using Fuzzy Logic

Table 6.1: Load forecasting results using Fuzzy Logic

TIME (24 HOUR)	ACTUAL LOAD	TRIANGULAR FORECAST	GAUSSIAN FORECAST	TRIANGULAR FORECAST ERROR (%)	GAUSSIAN FORECAST ERROR (%)
00:00:00	186.65	179.7562	180.5955	-3.69344	-3.2438
00:15:00	187.52	179.0848	179.4205	-4.49829	-4.31927
00:30:00	185.53	178.5813	178.5813	-3.74535	-3.74535
00:45:00	182.97	178.5813	178.4134	-2.39862	-2.49035
01:00:00	180.58	178.4134	178.5813	-1.1998	-1.10685
01:15:00	179.13	177.742	177.742	-0.77486	-0.77486
01:30:00	175.88	171.1959	171.6994	-2.66326	-2.37696
01:45:00	170.48	169.8531	169.3495	-0.36776	-0.66313
02:00:00	164.17	165.1533	166.8318	0.598922	1.621338
02:15:00	164.76	165.9925	165.9925	0.748058	0.748058
02:30:00	163.27	165.3211	165.1533	1.256263	1.153457
02:45:00	160.33	162.132	161.7963	1.123901	0.91452
03:00:00	156.94	160.957	159.4464	2.559577	1.597012
03:15:00	152.82	154.243	154.0752	0.931161	0.821326
03:30:00	149.94	155.9215	153.4038	3.989262	2.310091

03:45:00	149.57	154.243	152.2288	3.12429	1.777629
04:00:00	148.77	147.8647	145.8505	-0.60852	-1.96243
04:15:00	145.55	138.633	144.8434	-4.75235	-0.48547
04:30:00	142.17	134.7724	139.8079	-5.20335	-1.66146
04:45:00	138.5	129.7369	136.6188	-6.32715	-1.3583
05:00:00	133.81	128.0584	136.4509	-4.29833	1.973619
05:15:00	129.64	124.03	128.8977	-4.32737	-0.57262
05:30:00	123.03	126.0442	123.6943	2.449972	0.53995
05:45:00	115.81	110.602	114.4626	-4.49702	-1.1635
06:00:00	108.85	111.777	104.7273	2.688976	-3.78755
06:15:00	102.47	107.0772	101.5381	4.496096	-0.90944
06:30:00	98.2	98.68465	94.1527	0.493534	-4.12149
06:45:00	94.06	95.66335	91.29925	1.704603	-2.93509
07:00:00	90.84	94.1527	90.1243	3.646742	-0.78787
07:15:00	87.03	85.7602	86.26375	-1.45904	-0.88044
07:30:00	83.56	82.7389	81.56395	-0.98265	-2.38876
07:45:00	82.25	82.7389	80.22115	0.594407	-2.46669
08:00:00	80.98	82.7389	79.0462	2.172018	-2.388
08:15:00	77.38	82.7389	77.87125	6.925433	0.634854
08:30:00	76.11	82.0675	77.87125	7.827487	2.314085
08:45:00	76.68	82.0675	77.87125	7.025952	1.553534
09:00:00	75.58	77.53555	77.7034	2.587391	2.809473

09:15:00	75.08	77.53555	77.7034	3.270578	3.49414
09:30:00	74.93	77.53555	77.7034	3.477312	3.701321
09:45:00	73.58	77.53555	77.7034	5.375849	5.603968
10:00:00	72.53	77.53555	77.7034	6.901351	7.132773
10:15:00	71.74	77.3677	77.53555	7.844578	8.078548
10:30:00	71.27	77.032	76.0249	8.084748	6.671671
10:45:00	72.75	77.7034	77.87125	6.808797	7.039519
11:00:00	72.89	78.0391	77.87125	7.064206	6.833928
11:15:00	73.6	78.0391	77.7034	6.031386	5.575272
11:30:00	75.76	78.0391	77.19985	3.008316	1.900541
11:45:00	74.81	77.87125	77.19985	4.092033	3.19456
12:00:00	76.97	77.87125	77.87125	1.170911	1.170911
12:15:00	80.18	77.53555	77.87125	-3.29814	-2.87946
12:30:00	83.24	77.19985	77.87125	-7.25631	-6.44972
12:45:00	81.67	77.3677	77.87125	-5.26791	-4.65134
13:00:00	83.58	78.20695	78.7105	-6.42863	-5.82615
13:15:00	86.45	85.59235	85.7602	-0.99208	-0.79792
13:30:00	89.9	90.29215	90.29215	0.436207	0.436207
13:45:00	92.61	90.96355	91.1314	-1.77783	-1.59659
14:00:00	90.65	94.4884	94.1527	4.234308	3.863982
14:15:00	91.2	94.4884	94.32055	3.605702	3.421656
14:30:00	93.69	99.35605	98.8525	6.047657	5.510193

14:45:00	98.28	106.4058	105.7344	8.267959	7.584809
15:00:00	100.64	109.0914	106.4058	8.397605	5.729084
15:15:00	104.32	109.9306	109.4271	5.378259	4.895562
15:30:00	108.71	113.4555	113.1198	4.365238	4.056435
15:45:00	109.14	111.9448	111.777	2.56991	2.416117
16:00:00	110.4	112.4484	112.2805	1.855389	1.703351
16:15:00	108.81	112.1127	111.9448	3.035245	2.880985
16:30:00	106.61	112.2805	111.777	5.318919	4.84659
16:45:00	105.56	107.7486	107.5807	2.073276	1.914267
17:00:00	102.43	106.2379	105.9022	3.717563	3.389827
17:15:00	98.56	93.4813	101.3703	-5.1529	2.851309
17:30:00	91.41	92.8099	92.8099	1.531452	1.531452
17:45:00	86.92	90.96355	90.62785	4.652036	4.265819
18:00:00	81.51	87.4387	86.93515	7.273586	6.655809
18:15:00	74.26	76.52845	76.3606	3.05474	2.82871
18:30:00	70.85	75.3535	73.84285	6.356387	4.224206
18:45:00	72.92	77.87125	77.53555	6.789975	6.329608
19:00:00	73.6	78.0391	77.7034	6.031386	5.575272
19:15:00	73.2	78.3748	78.0391	7.069399	6.610792
19:30:00	74.46	76.52845	76.3606	2.777934	2.552511
19:45:00	76.96	84.7531	79.54975	10.12617	3.36506
20:00:00	82.56	87.103	86.7673	5.502665	5.096051

20:15:00	87.41	93.98485	90.7957	7.521851	3.873355
20:30:00	93.66	98.01325	97.67755	4.647929	4.289505
20:45:00	97.59	103.5523	103.0488	6.10954	5.593555
21:00:00	102.53	104.8951	104.7273	2.306739	2.143031
21:15:00	110.98	116.3089	115.9732	4.801676	4.499189
21:30:00	118.69	121.848	121.6801	2.660671	2.519252
21:45:00	125.27	131.919	131.4154	5.307695	4.905724
22:00:00	132.9	141.4864	140.815	6.460798	5.955606
22:15:00	140.08	153.2359	147.529	9.391705	5.317676
22:30:00	150.33	158.4393	158.9428	5.394299	5.729262
22:45:00	160.74	170.5245	170.0209	6.087128	5.773858
23:00:00	167.57	181.099	170.1888	8.073641	1.56278
23:15:00	177.32	182.7775	173.2101	3.077769	-2.31782
23:30:00	184.11	182.7775	178.5813	-0.72375	-3.00296
23:45:00	190.2	182.7775	181.9383	-3.90247	-4.34372

The above reading was obtained with the help of fuzzy logic and the following graph was obtained:-

(a) With triangular function

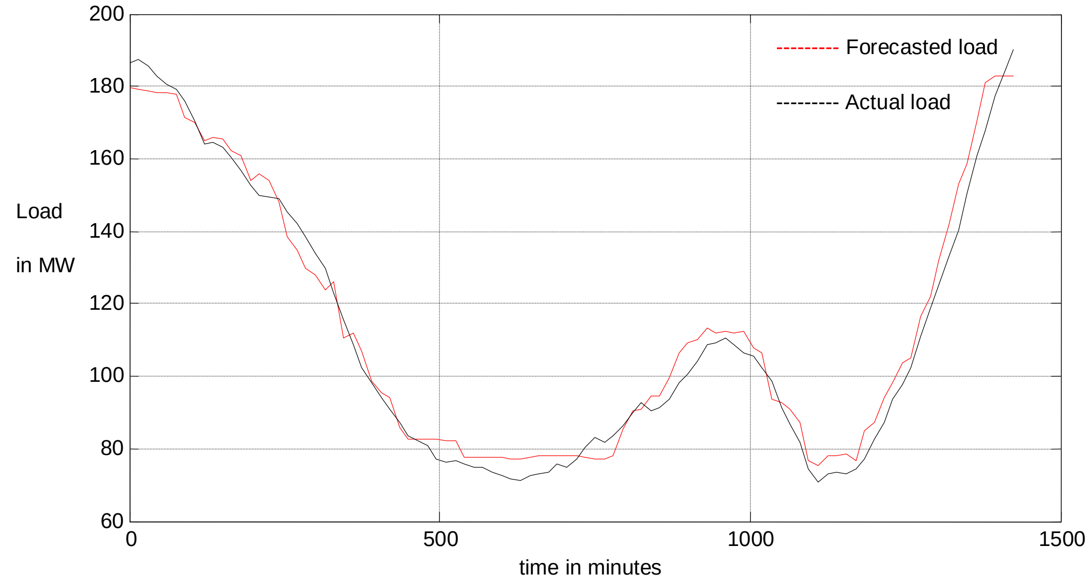

Fig 6.1 : Actual and predicted load curves with triangular membership function

(b) With gaussian function

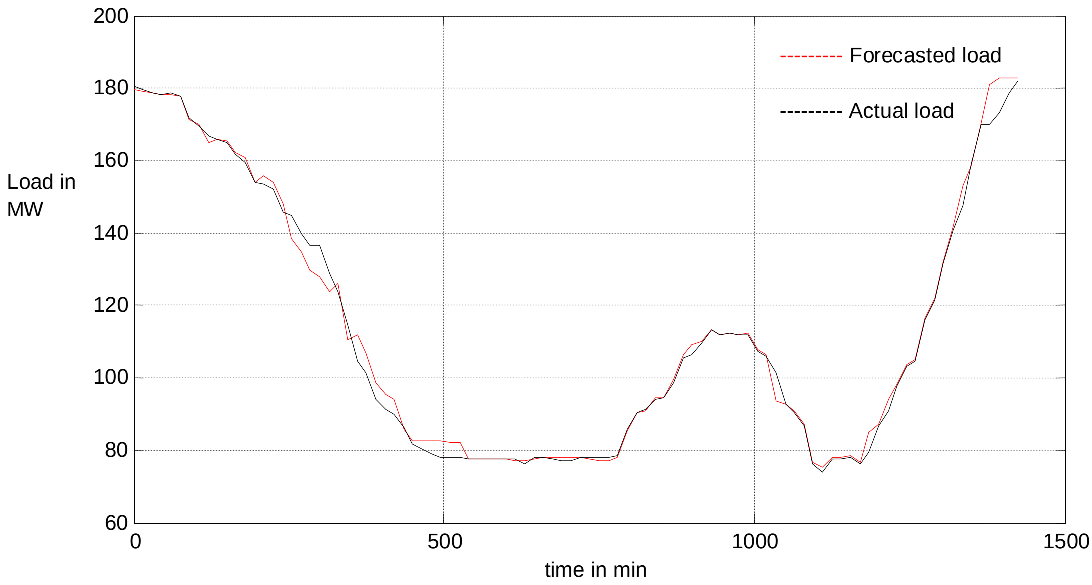

Fig 6.2 : Actual and predicted load curves with gaussian membership function

From Table 6.1, it can be observed that Based on the quarter-hourly load data for four days of the week, the load data for the fifth day was predicted using fuzzy logic. Both triangular and Gaussian membership functions were used. The rule base comprised forty eight rule bases.

It is observed that the predicted load curve closely matches the actual load data for the fifth day. The maximum errors with the triangular and Gaussian membership functions were observed to be 9.32% and 8.07%, respectively. The MAPE errors with the triangular and Gaussian membership functions were observed to be 2.35% and 1.66%, respectively.

CASE II: Load Forecasting using ANN

Table 6.2: Load forecasting results using ANN

TIME	ACTUAL LOAD	FORECASTED LOAD	PERCENTAGE ERROR
00:00:00	186.65	187.0517	-0.21522
00:15:00	187.52	186.3396	0.62948
00:30:00	185.53	184.0694	0.787258
00:45:00	182.97	183.3999	-0.23496
01:00:00	180.58	181.8762	-0.7178
01:15:00	179.13	178.6041	0.293586
01:30:00	175.88	177.3166	-0.81681
01:45:00	170.48	172.4337	-1.146
02:00:00	164.17	168.7432	-2.78565
02:15:00	164.76	164.0880	0.407866
02:30:00	163.27	161.0298	1.372083
02:45:00	160.33	158.8732	0.908626

03:00:00	156.94	156.5772	0.231171
03:15:00	152.82	156.0196	-2.09371
03:30:00	149.94	152.8036	-1.90983
03:45:00	149.57	150.6541	-0.72481
04:00:00	148.77	147.9341	0.561874
04:15:00	145.55	144.1783	0.942425
04:30:00	142.17	143.5880	-0.9974
04:45:00	138.5	140.2615	-1.27184
05:00:00	133.81	137.8257	-3.00105
05:15:00	129.64	128.8637	0.598812
05:30:00	123.03	126.6189	-2.91709
05:45:00	115.81	115.7291	0.069856
06:00:00	108.85	108.3735	0.437758
06:15:00	102.47	103.0820	-0.59725
06:30:00	98.2	98.5639	-0.37057
06:45:00	94.06	95.2911	-1.30885
07:00:00	90.84	92.9953	-2.37263
07:15:00	87.03	88.1629	-1.30174
07:30:00	83.56	85.9865	-2.9039
07:45:00	82.25	84.1905	-2.35927
08:00:00	80.98	81.9407	-1.18634
08:15:00	77.38	78.3516	-1.25562

08:30:00	76.11	76.6818	-0.75128
08:45:00	76.68	75.7368	1.230047
09:00:00	75.58	74.5855	1.315824
09:15:00	75.08	75.0203	0.079515
09:30:00	74.93	75.0744	-0.19271
09:45:00	73.58	74.7245	-1.55545
10:00:00	72.53	73.1860	-0.90445
10:15:00	71.74	74.3446	-3.63061
10:30:00	71.27	75.3262	-5.69131
10:45:00	72.75	74.0180	-1.74296
11:00:00	72.89	73.2662	-0.51612
11:15:00	73.6	73.5423	0.078397
11:30:00	75.76	73.5683	2.892951
11:45:00	74.81	74.4543	0.475471
12:00:00	76.97	75.7937	1.528258
12:15:00	80.18	76.2458	4.90671
12:30:00	83.24	79.2557	4.786521
12:45:00	81.67	80.3362	1.633158
13:00:00	83.58	81.7894	2.142379
13:15:00	86.45	84.9750	1.706189
13:30:00	89.9	87.9601	2.157842
13:45:00	92.61	88.9665	3.93424

14:00:00	90.65	91.8829	-1.36007
14:15:00	91.2	92.3917	-1.30669
14:30:00	93.69	96.5051	-3.0047
14:45:00	98.28	101.0118	-2.77961
15:00:00	100.64	102.2400	-1.58983
15:15:00	104.32	104.5709	-0.24051
15:30:00	108.71	109.3836	-0.61963
15:45:00	109.14	109.3031	-0.14944
16:00:00	110.4	110.0574	0.310326
16:15:00	108.81	104.1852	4.250345
16:30:00	106.61	108.8467	-2.09802
16:45:00	105.56	106.3656	-0.76317
17:00:00	102.43	99.3953	2.962706
17:15:00	98.56	91.0756	7.59375
17:30:00	91.41	91.9613	-0.60311
17:45:00	86.92	86.9442	-0.02784
18:00:00	81.51	84.4636	-3.6236
18:15:00	74.26	79.7033	-7.33006
18:30:00	70.85	73.7106	-4.03754
18:45:00	72.92	73.7286	-1.10889
19:00:00	73.6	74.0549	-0.61807
19:15:00	73.2	73.8741	-0.9209

19:30:00	74.46	76.8926	-3.26699
19:45:00	76.96	77.6395	-0.88293
20:00:00	82.56	81.2757	1.555596
20:15:00	87.41	86.9843	0.487015
20:30:00	93.66	90.3171	3.569186
20:45:00	97.59	96.6797	0.93278
21:00:00	102.53	104.7526	-2.16776
21:15:00	110.98	109.8829	0.988556
21:30:00	118.69	117.8552	0.703345
21:45:00	125.27	126.7579	-1.18775
22:00:00	132.9	134.4888	-1.19549
22:15:00	140.08	144.5137	-3.16512
22:30:00	150.33	150.8081	-0.31803
22:45:00	160.74	159.8771	0.53683
23:00:00	167.57	167.1252	0.265441
23:15:00	177.32	176.4272	0.503497
23:30:00	184.11	183.8373	0.148118
23:45:00	190.2	187.4375	1.452419

From Table 6.2 the following inferences can be drawn

1) Regression plot

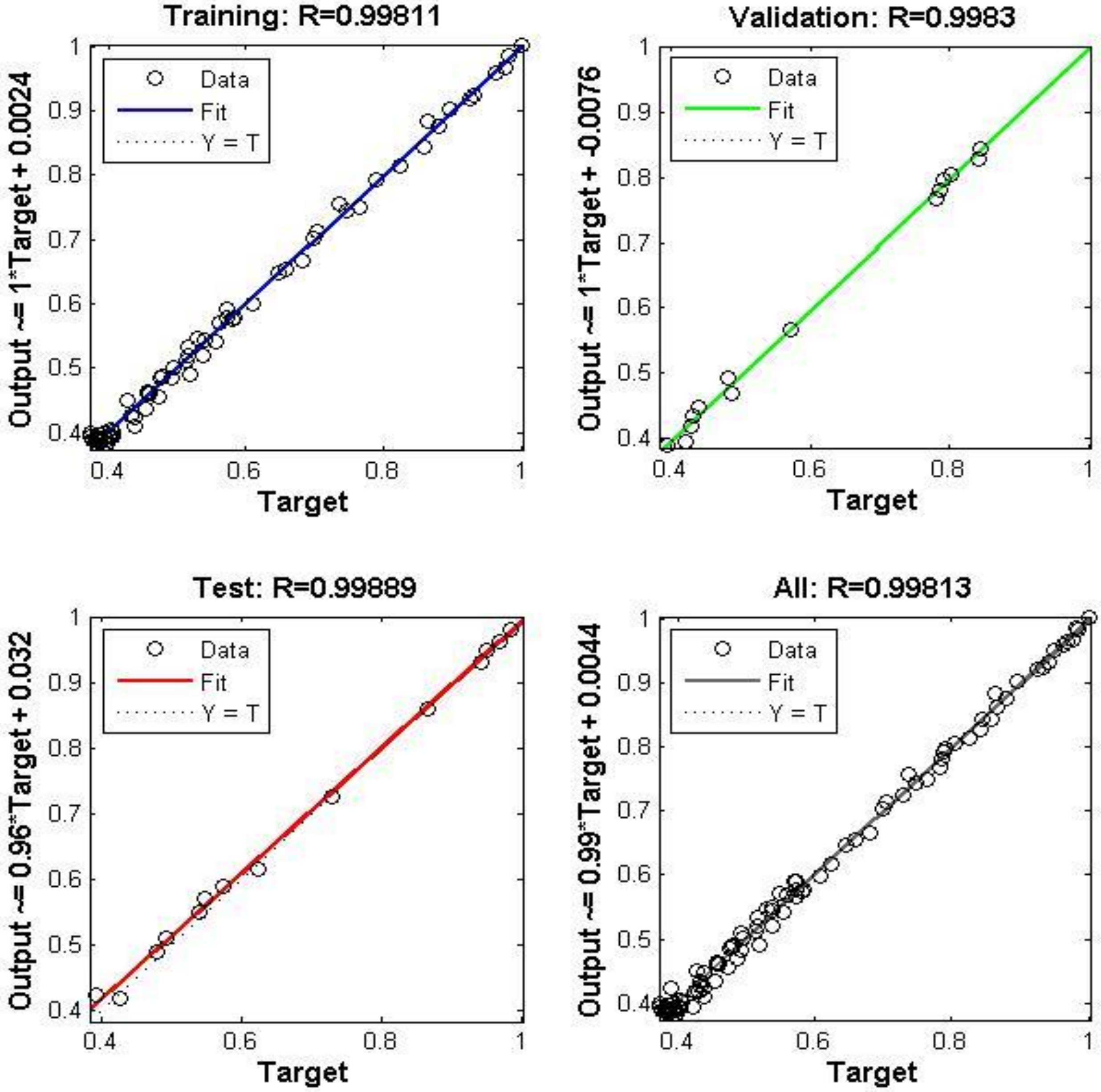

Fig 6.3: Regression plot of the neural network

2) Training performance

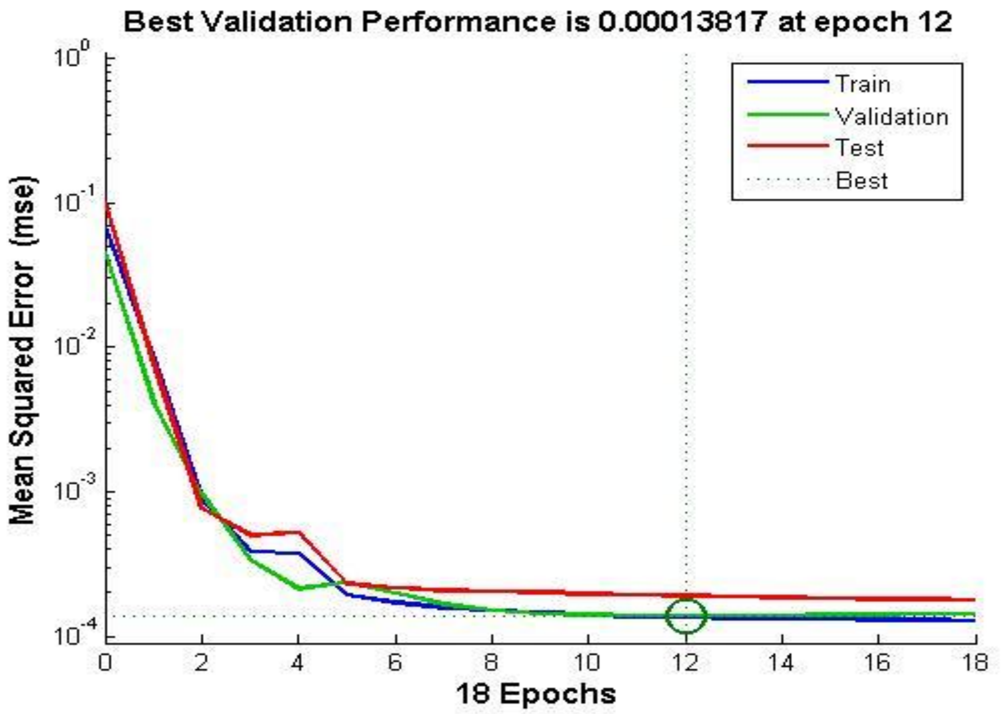

Fig 6.4: Training performance of neural trained network

3) Actual and Forecasted Load

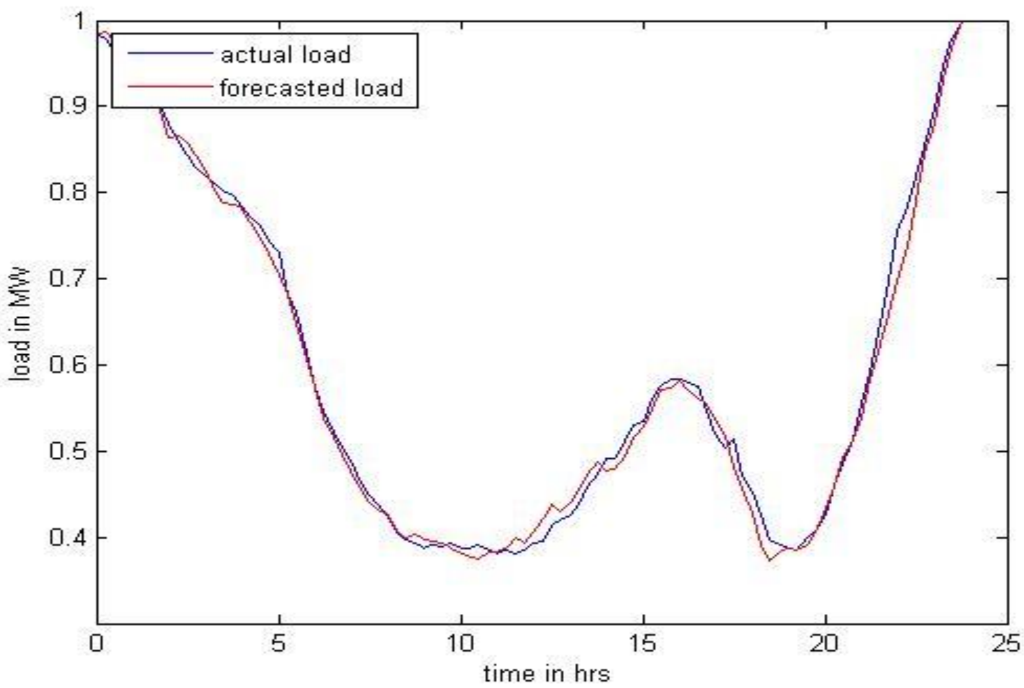

Fig 6.5: Actual and forecasted load using ANN

Based on the quarter-hourly load data for four days of the week, the load data for the fifth day was predicted using Artificial Neural Network. The training was performed with the all possible no of neurons.

It is observed that the predicted load curve closely matches the actual load data for the fifth day. The errors with the different neurons were observed from 1.4% to 2.5%, min and max respectively.

<u>Case III: Load forecasting Using WNN</u>

Table 6.3: Load forecasting results using WNN

TIME (24 HOUR)	**Forecasted output**	**Actual output**	**Gross error**
00:00:00	182.58	186.65	2.1828
00:15:00	184.94	187.52	1.3773
00:30:00	186.02	185.53	-0.2652
00:45:00	185.89	182.97	-1.5978
01:00:00	184.73	180.58	-2.2960
01:15:00	182.88	179.13	-2.0908
01:30:00	180.60	175.88	-2.6840
01:45:00	177.94	170.48	-4.3764
02:00:00	174.91	164.17	-6.5406
02:15:00	171.40	164.76	-4.0277
02:30:00	167.36	163.27	-2.5038
02:45:00	163.05	160.33	-1.6936
03:00:00	158.61	156.94	-1.0656
03:15:00	154.11	152.82	-0.8413

03:30:00	149.65	149.94	0.1959
03:45:00	145.25	149.57	2.8855
04:00:00	141.00	148.77	5.2243
04:15:00	137.09	145.55	5.8110
04:30:00	133.60	142.17	6.0278
04:45:00	130.41	138.50	5.8399
05:00:00	127.38	133.81	4.8073
05:15:00	124.21	129.64	4.1861
05:30:00	120.72	123.03	1.8812
05:45:00	116.95	115.81	-0.9854
06:00:00	112.98	108.85	-3.7933
06:15:00	108.90	102.47	-6.2733
06:30:00	104.84	98.20	-6.7579
06:45:00	100.75	94.06	-7.1151
07:00:00	96.69	90.84	-6.4396
07:15:00	92.81	87.03	-6.6389
07:30:00	89.17	83.56	-6.7158
07:45:00	85.82	82.25	-4.3461
08:00:00	82.77	80.98	-2.2126
08:15:00	79.88	77.38	-3.2249
08:30:00	77.07	76.11	-1.2640
08:45:00	74.48	76.68	2.8742

09:00:00	72.18	75.58	4.4988
09:15:00	70.31	75.08	6.3595
09:30:00	68.98	74.93	7.9444
09:45:00	68.15	73.58	7.3735
10:00:00	67.81	72.53	6.5012
10:15:00	67.99	71.74	5.2336
10:30:00	68.62	71.27	3.7143
10:45:00	69.69	72.75	4.2016
11:00:00	71.13	72.89	2.4183
11:15:00	72.76	73.60	1.1465
11:30:00	74.47	75.76	1.7026
11:45:00	76.25	74.81	-1.9305
12:00:00	78.10	76.97	-1.4708
12:15:00	80.07	80.18	0.1384
12:30:00	82.20	83.24	1.2546
12:45:00	84.41	81.67	-3.3562
13:00:00	86.68	83.58	-3.7090
13:15:00	89.00	86.45	-2.9538
13:30:00	91.35	89.90	-1.6080
13:45:00	93.69	92.61	-1.1670
14:00:00	95.99	90.65	-5.8924
14:15:00	98.12	91.20	-7.5856

14:30:00	99.97	93.69	-6.7073
14:45:00	101.52	98.28	-3.2988
15:00:00	102.71	100.64	-2.0579
15:15:00	103.51	104.32	0.7770
15:30:00	103.90	108.71	4.4253
15:45:00	103.83	109.14	4.8639
16:00:00	103.29	110.40	6.4407
16:15:00	102.31	108.81	5.9774
16:30:00	100.90	106.61	5.3515
16:45:00	99.11	105.56	6.1104
17:00:00	96.97	102.43	5.3297
17:15:00	94.55	98.56	4.0729
17:30:00	91.91	91.41	-0.5471
17:45:00	89.16	86.92	-2.5819
18:00:00	86.41	81.51	-6.0078
18:15:00	83.73	74.26	-12.7541
18:30:00	81.26	70.85	-14.6974
18:45:00	79.18	72.92	-8.5831
19:00:00	77.68	73.60	-5.5430
19:15:00	77.00	73.20	-5.1906
19:30:00	77.33	74.46	-3.8542
19:45:00	78.74	76.96	-2.3179

20:00:00	81.27	82.56	1.5631
20:15:00	84.88	87.41	2.8965
20:30:00	89.51	93.66	4.4300
20:45:00	95.14	97.59	2.5124
21:00:00	101.62	102.53	0.8861
21:15:00	108.71	110.98	2.0478
21:30:00	116.23	118.69	2.0763
21:45:00	124.04	125.27	0.9848
22:00:00	132.02	132.90	0.6623
22:15:00	140.22	140.08	-0.1005
22:30:00	148.55	150.33	1.1831
22:45:00	156.73	160.74	2.4967
23:00:00	164.51	167.57	1.8263
23:15:00	171.63	177.32	3.2065
23:30:00	177.89	184.11	3.3771
23:45:00	183.18	190.20	3.6932

From the Table 6.3 the following inferences can be drawn from Wavelet neural application

1) Regression Plot

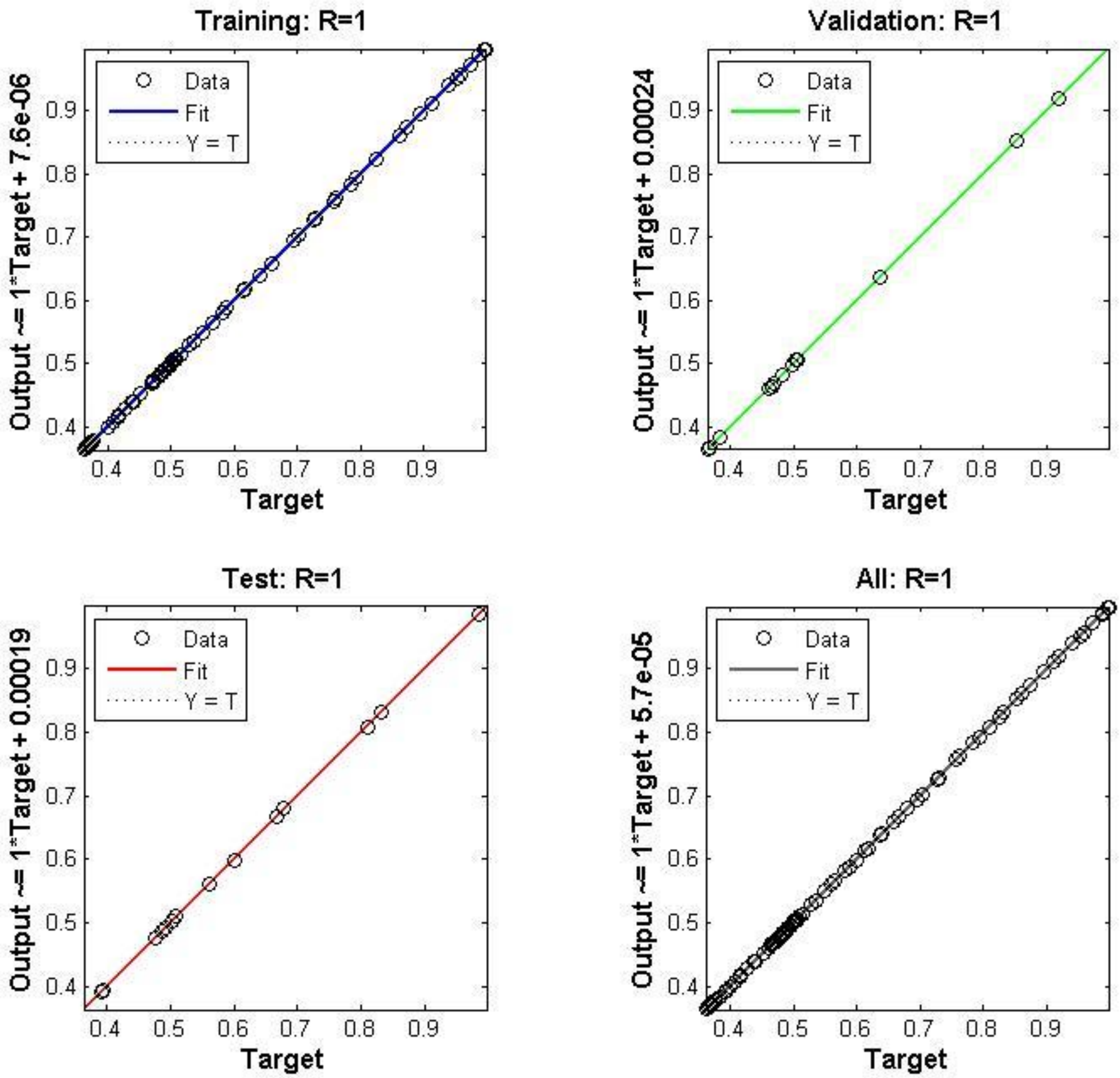

Fig 6.6: Regression plot using WNN

2) Training performance

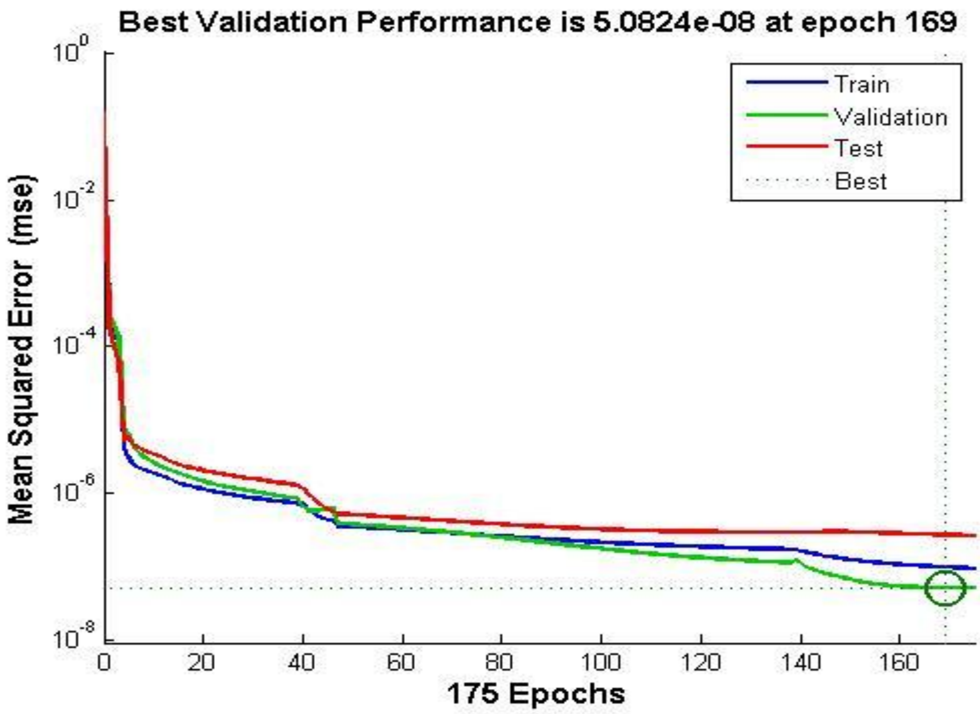

Fig 6.7: Training performance of the wavelet neural network

3) Actual and Forecasted Load

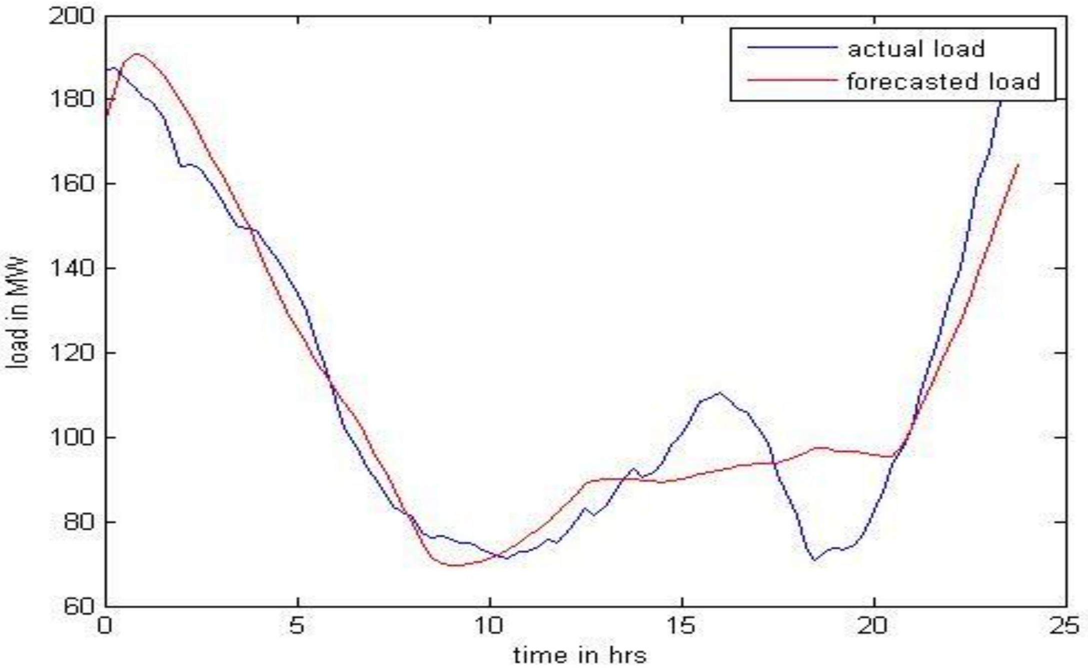

Fig 6.8: Actual and forecasted load using WNN

Based on the quarter-hourly load data for four days of the week, the load data for the fifth day was predicted using Wavelet Neural Network. The training was performed with different wavelet families.

It is observed that the predicted load curve matches the actual load data for the fifth day with small variations. The minimum error was observed upto 3.77% from db8.

Case IV: Load Forecasting using WNN Spike Filtering

Table 6.4: Load forecasting result data using Data pre-filtering

Time in Hrs	Actual load (MW)	Forecasted load (MW)	Percentage error (%)
00:00	5445	5428.0313	0.3116
00:05	5454	5413.8841	0.7355
00:10	5441	5400.6046	0.7424
00:15	5344	5388.2691	-0.8284
00:20	5319	5376.9076	-1.0887
00:25	5282	5366.5254	-1.6003
00:30	5391	5357.1946	0.6271
00:35	5340	5348.8713	-0.1661
00:40	5313	5341.5573	-0.5375
00:45	5271	5335.3948	-1.2217
00:50	5325	5330.4082	-0.1016
00:55	5354	5326.4903	0.5138
01:00	5343	5323.4860	0.3652
01:05	5343	5321.0783	0.4103
01:10	5319	5318.9639	0.0007

01:15	5297	5317.0412	-0.3784
01:20	5336	5315.2108	0.3896
01:25	5294	5313.4269	-0.3670
01:30	5320	5311.7083	0.1559
01:35	5277	5309.9641	-0.6247
01:40	5346	5308.2147	0.7068
01:45	5312	5306.6549	0.1006
01:50	5335	5305.4099	0.5546
01:55	5338	5304.5408	0.6268
02:00	5326	5304.0756	0.4116
02:05	5321	5303.8984	0.3214
02:10	5256	5303.9712	-0.9127
02:15	5341	5304.4724	0.6839
02:20	5319	5305.5176	0.2535
02:25	5277	5307.1805	-0.5719
02:30	5256	5309.5324	-1.0185
02:35	5338	5312.5112	0.4775
02:40	5358	5316.0884	0.7822
02:45	5317	5320.3295	-0.0626
02:50	5331	5325.2308	0.1082
02:55	5348	5330.7329	0.3229
03:00	5332	5336.7599	-0.0893
03:05	5357	5343.1537	0.2585

03:10	5342	5349.8052	-0.1461
03:15	5309	5356.7580	-0.8996
03:20	5331	5364.0636	-0.6202
03:25	5313	5371.8074	-1.1069
03:30	5342	5380.0888	-0.7130
03:35	5414	5388.9130	0.4634
03:40	5413	5398.3450	0.2707
03:45	5406	5408.5441	-0.0471
03:50	5373	5419.6428	-0.8681
03:55	5453	5431.7707	0.3893
04:00	5457	5445.0558	0.2189
04:05	5494	5459.5790	0.6265
04:10	5551	5475.4797	1.3605
04:15	5477	5493.0009	-0.2921
04:20	5473	5512.3494	-0.7190
04:25	5506	5533.6848	-0.5028
04:30	5536	5557.1674	-0.3824
04:35	5549	5582.9318	-0.6115
04:40	5590	5611.1295	-0.3780
04:45	5654	5641.9492	0.2131
04:50	5732	5675.5120	0.9855
04:55	5750	5711.9276	0.6621
05:00	5808	5751.3647	0.9751

05:05	5911	5793.9712	1.9798
05:10	5946	5839.9176	1.7841
05:15	5928	5889.3312	0.6523
05:20	5945	5942.2464	0.0463
05:25	5950	5998.7256	-0.8189
05:30	5995	6058.6542	-1.0618
05:35	6030	6121.5732	-1.5186
05:40	6211	6186.9181	0.3877
05:45	6198	6254.0172	-0.9038
05:50	6300	6322.3145	-0.3542
05:55	6306	6391.6033	-1.3575
06:00	6421	6461.5376	-0.6313
06:05	6528	6531.4662	-0.0531
06:10	6563	6600.5032	-0.5714
06:15	6686	6667.6889	0.2739
06:20	6721	6732.2446	-0.1673
06:25	6686	6793.4641	-1.6073
06:30	6890	6850.6913	0.5705
06:35	6928	6903.3198	0.3562
06:40	6972	6950.7667	0.3046
06:45	6965	6992.7579	-0.3985
06:50	7079	7029.2169	0.7033
06:55	7102	7060.0971	0.5900

07:00	7107	7085.7204	0.2994
07:05	7173	7106.6731	0.9247
07:10	7112	7123.4281	-0.1607
07:15	7126	7136.3588	-0.1454
07:20	7186	7145.6873	0.5610
07:25	7207	7151.3001	0.7729
07:30	7184	7153.1976	0.4288
07:35	7099	7151.8436	-0.7444
07:40	7179	7147.8176	0.4344
07:45	7170	7141.9923	0.3906
07:50	7193	7135.3250	0.8018
07:55	7162	7128.2200	0.4717
08:00	7130	7120.9366	0.1271
08:05	7039	7113.4545	-1.0577
08:10	7023	7105.3178	-1.1721
08:15	7091	7096.1314	-0.0724
08:20	7093	7085.3394	0.1080
08:25	7047	7072.3666	-0.3600
08:30	7006	7057.2269	-0.7312
08:35	7023	7040.2496	-0.2456
08:40	7020	7021.9077	-0.0272
08:45	6980	7002.7567	-0.3260
08:50	6984	6982.9432	0.0151

08:55	6917	6962.6863	-0.6605
09:00	6869	6942.2092	-1.0658
09:05	6931	6921.2894	0.1401
09:10	6914	6899.7533	0.2061
09:15	6879	6877.2689	0.0252
09:20	6881	6853.7196	0.3965
09:25	6856	6830.2670	0.3753
09:30	6851	6808.2415	0.6241
09:35	6843	6788.1199	0.8020
09:40	6709	6770.2006	-0.9122
09:45	6802	6753.5608	0.7121
09:50	6746	6736.8182	0.1361
09:55	6722	6720.3973	0.0238
10:00	6685	6704.4239	-0.2906
10:05	6691	6688.7360	0.0338
10:10	6717	6673.6046	0.6461
10:15	6689	6657.9794	0.4638
10:20	6643	6641.5962	0.0211
10:25	6669	6625.8821	0.6465
10:30	6631	6611.2479	0.2979
10:35	6599	6597.1473	0.0281
10:40	6585	6582.6945	0.0350
10:45	6529	6565.7896	-0.5635

10:50	6535	6545.4111	-0.1593
10:55	6508	6522.7150	-0.2261
11:00	6485	6498.5748	-0.2093
11:05	6464	6473.8877	-0.1530
11:10	6395	6449.6582	-0.8547
11:15	6420	6426.1334	-0.0955
11:20	6337	6403.6845	-1.0523
11:25	6397	6382.5357	0.2261
11:30	6406	6362.3093	0.6820
11:35	6299	6342.6821	-0.6935
11:40	6274	6323.4224	-0.7877
11:45	6324	6304.6132	0.3066
11:50	6326	6286.7037	0.6212
11:55	6274	6269.7697	0.0674
12:00	6277	6253.9308	0.3675
12:05	6274	6239.5237	0.5495
12:10	6251	6226.5911	0.3905
12:15	6239	6215.1852	0.3817
12:20	6220	6205.0847	0.2398
12:25	6134	6195.3776	-1.0006
12:30	6196	6185.4164	0.1708
12:35	6219	6175.2592	0.7033
12:40	6181	6165.0735	0.2577

12:45	6150	6155.5188	-0.0897
12:50	6148	6147.1655	0.0136
12:55	6174	6139.4683	0.5593
13:00	6047	6132.0441	-1.4064
13:05	6117	6124.8080	-0.1276
13:10	6103	6117.2749	-0.2339
13:15	6080	6109.2789	-0.4816
13:20	6119	6100.5021	0.3023
13:25	6025	6089.9229	-1.0776
13:30	6092	6077.1200	0.2443
13:35	6082	6062.6165	0.3187
13:40	6050	6047.1081	0.0478
13:45	6034	6031.8506	0.0356
13:50	6047	6017.9200	0.4809
13:55	6038	6005.2069	0.5431
14:00	5999	5993.6633	0.0890
14:05	5928	5983.3155	-0.9331
14:10	5967	5973.9624	-0.1167
14:15	5939	5965.8764	-0.4525
14:20	5974	5959.0014	0.2511
14:25	5976	5952.6431	0.3908
14:30	5960	5946.4373	0.2276
14:35	5963	5940.3580	0.3797

14:40	5938	5934.5156	0.0587
14:45	5958	5929.5781	0.4770
14:50	5930	5926.1185	0.0655
14:55	5857	5924.0579	-1.1449
15:00	5858	5923.4355	-1.1170
15:05	5888	5924.1483	-0.6139
15:10	5909	5925.8553	-0.2852
15:15	5962	5928.7347	0.5580
15:20	5979	5932.8560	0.7718
15:25	5992	5938.0861	0.8998
15:30	5987	5944.5406	0.7092
15:35	5986	5952.0459	0.5672
15:40	5971	5960.5020	0.1758
15:45	5981	5970.2487	0.1798
15:50	5991	5981.3608	0.1609
15:55	5915	5993.6795	-1.3302
16:00	5999	6007.0011	-0.1334
16:05	5978	6020.7254	-0.7147
16:10	6072	6034.5195	0.6173
16:15	6009	6048.7967	-0.6623
16:20	6101	6063.9867	0.6067
16:25	6090	6080.6028	0.1543
16:30	6139	6099.1214	0.6496

16:35	6136	6119.4976	0.2689
16:40	6183	6141.7139	0.6677
16:45	6112	6165.8272	-0.8807
16:50	6144	6191.6765	-0.7760
16:55	6223	6218.9948	0.0644
17:00	6165	6247.3392	-1.3356
17:05	6231	6276.1508	-0.7246
17:10	6273	6305.0138	-0.5103
17:15	6316	6333.7251	-0.2806
17:20	6349	6362.1706	-0.2074
17:25	6401	6390.2960	0.1672
17:30	6447	6418.1677	0.4472
17:35	6384	6445.9540	-0.9705
17:40	6479	6473.8658	0.0792
17:45	6576	6502.1063	1.1237
17:50	6590	6530.6101	0.9012
17:55	6567	6558.9318	0.1229
18:00	6645	6586.4203	0.8816
18:05	6641	6612.2056	0.4336
18:10	6687	6635.5104	0.7700
18:15	6716	6656.0041	0.8933
18:20	6647	6673.5230	-0.3990
18:25	6659	6688.0372	-0.4361

18:30	6736	6699.6874	0.5391
18:35	6741	6708.4938	0.4822
18:40	6736	6714.4603	0.3198
18:45	6716	6717.6936	-0.0252
18:50	6715	6718.0929	-0.0461
18:55	6684	6715.3412	-0.4689
19:00	6604	6709.1877	-1.5928
19:05	6674	6699.4308	-0.3810
19:10	6560	6686.1164	-1.9225
19:15	6588	6669.6933	-1.2400
19:20	6642	6650.7287	-0.1314
19:25	6646	6629.9288	0.2418
19:30	6649	6608.1351	0.6146
19:35	6609	6586.1402	0.3459
19:40	6575	6564.6944	0.1567
19:45	6600	6544.3603	0.8430
19:50	6584	6525.3184	0.8913
19:55	6527	6507.5204	0.2984
20:00	6477	6490.7910	-0.2129
20:05	6496	6474.8739	0.3252
20:10	6398	6459.5880	-0.9626
20:15	6403	6444.7458	-0.6520
20:20	6424	6430.1787	-0.0962

20:25	6436	6415.8554	0.3130
20:30	6419	6401.7867	0.2682
20:35	6410	6387.9918	0.3433
20:40	6395	6374.4531	0.3213
20:45	6375	6360.9788	0.2199
20:50	6374	6347.3642	0.4179
20:55	6371	6333.5206	0.5883
21:00	6319	6319.2708	-0.0043
21:05	6217	6304.3771	-1.4055
21:10	6268	6288.5127	-0.3273
21:15	6193	6271.1803	-1.2624
21:20	6265	6251.9265	0.2087
21:25	6251	6230.3709	0.3300
21:30	6153	6206.3159	-0.8665
21:35	6191	6180.0090	0.1775
21:40	6174	6152.0701	0.3552
21:45	6160	6123.3883	0.5943
21:50	6124	6094.8430	0.4761
21:55	6134	6066.9312	1.0934
22:00	6008	6039.8809	-0.5306
22:05	5968	6013.7417	-0.7664
22:10	5965	5988.3355	-0.3912
22:15	6010	5963.3529	0.7762

22:20	5985	5938.4382	0.7780
22:25	5890	5913.2080	-0.3940
22:30	5903	5887.4193	0.2639
22:35	5883	5861.0534	0.3731
22:40	5843	5834.2236	0.1502
22:45	5827	5807.2564	0.3388
22:50	5743	5780.5684	-0.6542
22:55	5709	5754.4183	-0.7956
23:00	5722	5728.9976	-0.1223
23:05	5727	5704.3562	0.3954
23:10	5685	5680.4505	0.0800
23:15	5649	5657.4137	-0.1489
23:20	5615	5635.3322	-0.3621
23:25	5624	5614.2248	0.1738
23:30	5602	5594.1300	0.1405
23:35	5592	5574.9047	0.3057
23:40	5475	5556.4442	-1.4876
23:45	5490	5538.8046	-0.8890
23:50	5517	5521.9923	-0.0905
23:55	5486	5506.0153	-0.3648

From the Table 6.4 following inferences can be drawn

1) Regression plot

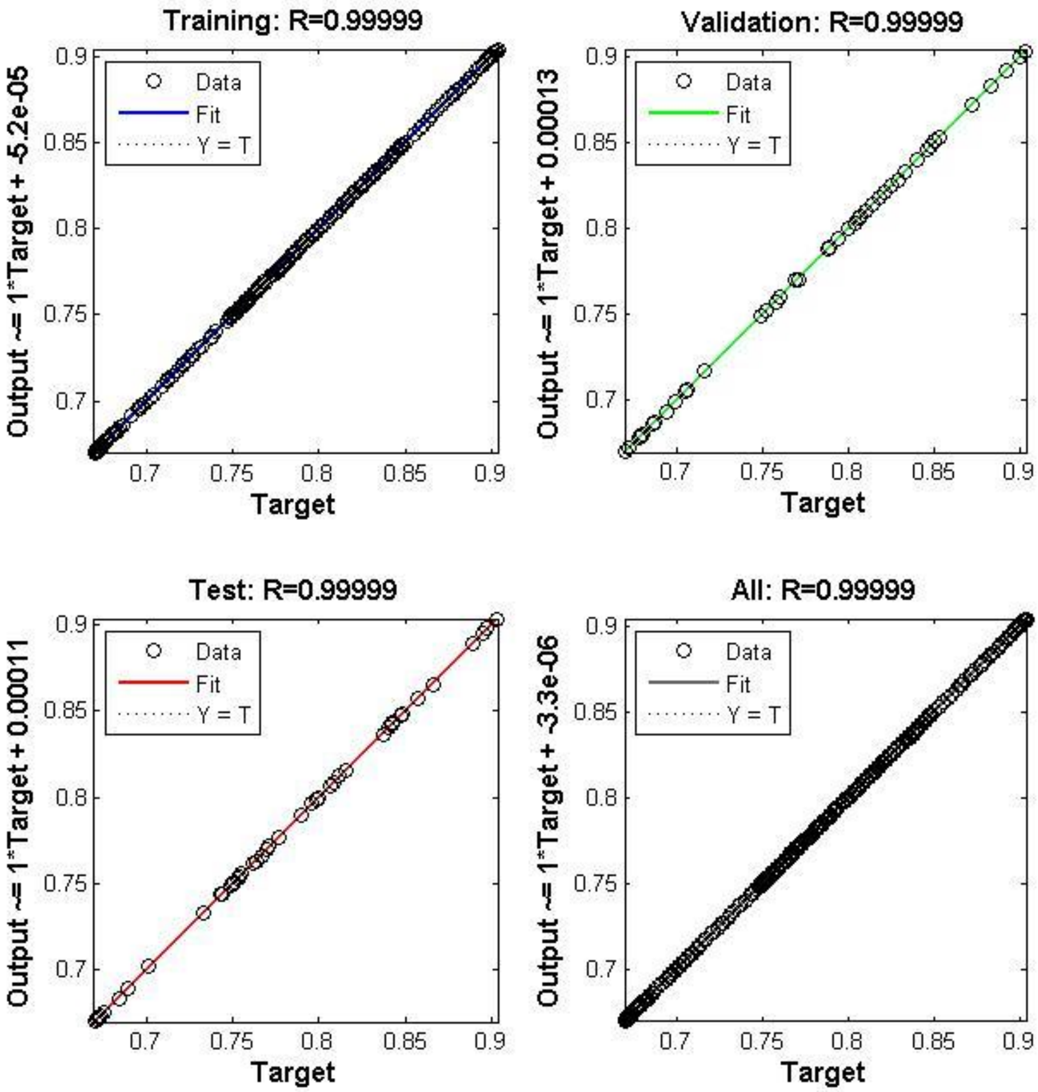

Fig 6.9: Regression plot of the spike filtered WNN

2) Training Performance

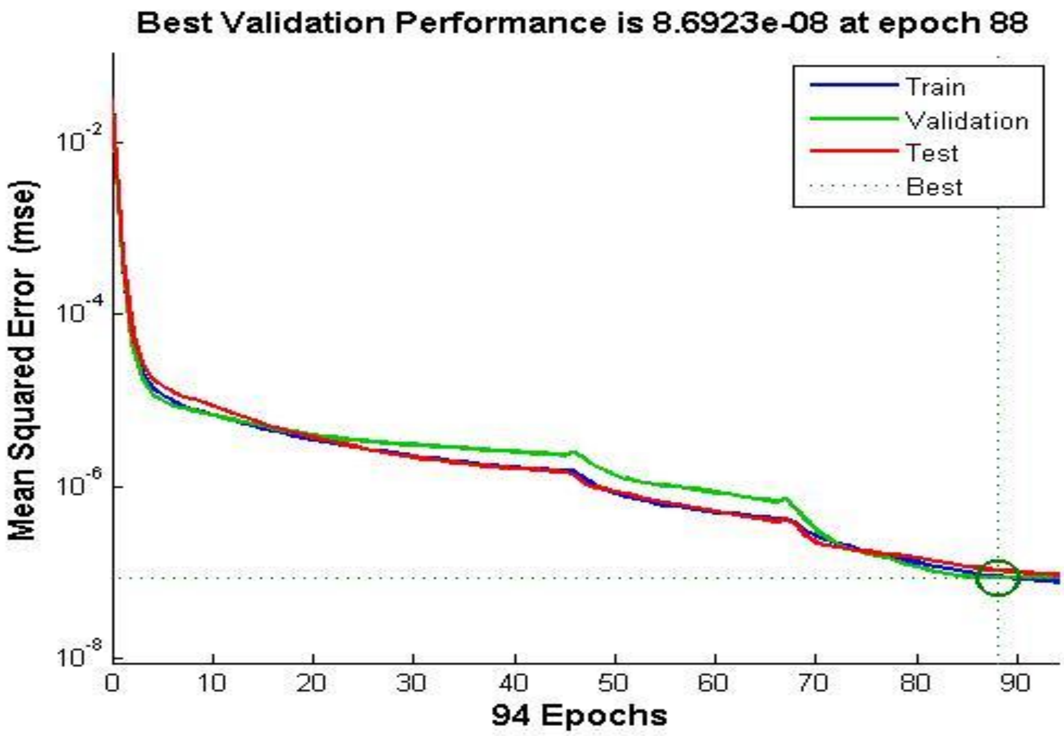

Fig 6.10: Training performance of spike filtered WNN

3) Actual and forecasted load

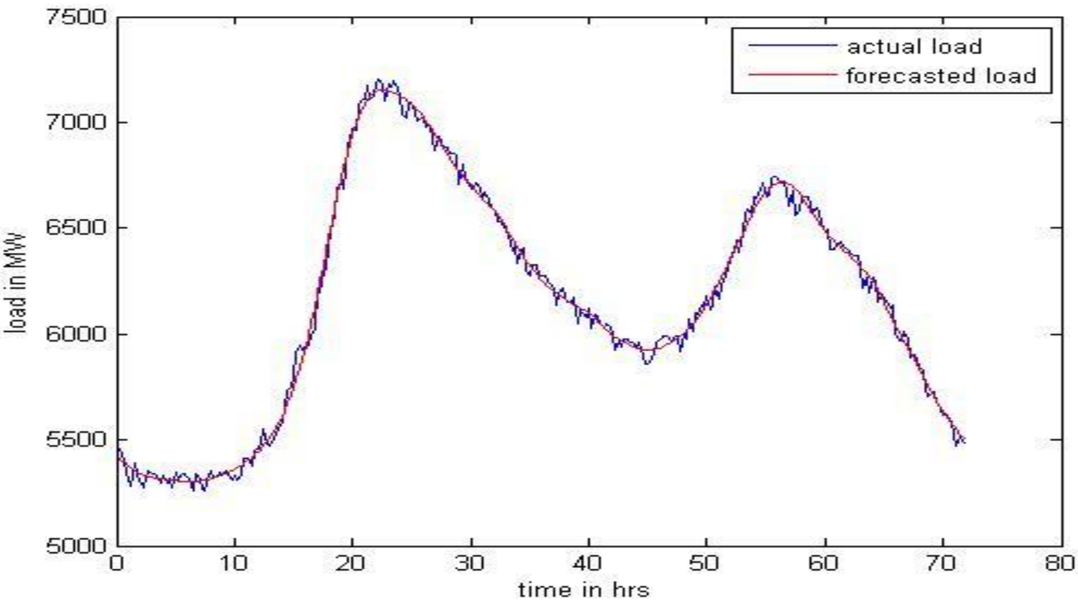

Fig 6.11: Actual and forecasted load using spike filtering WNN

Based on the 5 min load data for four days of the week, the load data for the fifth day was predicted using Wavelet Neural Network. The training was performed with different wavelet families.

It is observed that the predicted load curve matches the actual load data for the fifth day with very small variations. The minimum error was observed upto 0.5104% from db8.

CHAPTER 7: Conclusions

VSTLF was done with the help of three different Artificial Intelligence Techniques. Based on the observations, the maximum absolute percentage error was found to be approx 2.35% for triangular membership function and approx 1.66% for Gaussian membership function in case of forecasting through fuzzy logic whereas through Artificial Neural Network it was found to be approx 1.477% and by Wavelet Neural Network it was 3.77% approximately.

Results show that predicted result is best with ANN having the smallest deviation (error).

Daubechies-8 with two-level decomposition is the best configuration, which balances the decomposed level, the filter length, and the minimum padding length for decomposition. Based on test results, 12 dedicated wavelet neural networks are used to perform moving forecasts every 5 min. Numerical testing shows accurate predictions with small standard deviations for VSTLF of 0.5107% approximately.

Appendix

Table I: Load data of TPDDL from 11 April 2012 – 15 April 2012

Time (in Hrs)	Load on 11-04-2012	Load on 12-04-2012	Load on 13-04-2012	Load on 14-04-2012	Load on 15-04-2012
00:00:00	162.740	176.720	182.550	191.170	186.650
00:15:00	161.440	176.440	182.380	190.920	187.520
00:30:00	158.850	174.050	177.130	187.180	185.530
00:45:00	158.060	170.680	172.590	185.500	182.970
01:00:00	159.020	169.270	169.460	181.170	180.580
01:15:00	156.910	164.750	166.100	177.790	179.130
01:30:00	155.630	162.370	161.380	175.300	175.880
01:45:00	151.780	160.010	162.260	172.910	170.480
02:00:00	148.740	157.570	159.950	169.920	164.170
02:15:00	145.770	153.120	156.940	165.660	164.760
02:30:00	142.960	149.880	153.920	162.670	163.270
02:45:00	140.700	148.590	149.600	159.470	160.330
03:00:00	138.410	145.480	147.600	157.070	156.940
03:15:00	134.410	140.550	143.810	155.390	152.820
03:30:00	131.990	138.660	142.450	152.070	149.940
03:45:00	128.610	136.700	140.310	149.330	149.570
04:00:00	124.700	132.530	136.320	144.840	148.770

04:15:00	120.060	128.000	132.180	139.260	145.550
04:30:00	114.820	123.750	127.790	136.000	142.170
04:45:00	112.200	120.960	124.090	131.620	138.500
05:00:00	109.460	117.420	121.200	128.330	133.810
05:15:00	105.200	113.250	115.280	119.490	129.640
05:30:00	97.350	106.310	109.230	114.940	123.030
05:45:00	90.560	99.420	103.450	106.310	115.810
06:00:00	84.640	93.490	96.450	101.550	108.850
06:15:00	80.310	88.520	91.180	98.520	102.470
06:30:00	75.600	85.210	86.060	95.540	98.200
06:45:00	73.710	83.150	83.430	92.810	94.060
07:00:00	72.130	81.950	81.400	90.740	90.840
07:15:00	70.000	76.480	77.130	86.090	87.030
07:30:00	68.340	76.010	74.320	84.000	83.560
07:45:00	67.660	72.990	73.900	81.180	82.250
08:00:00	66.850	71.600	71.640	78.060	80.980
08:15:00	63.650	71.060	69.880	73.460	77.380
08:30:00	62.260	70.560	68.520	71.520	76.110
08:45:00	59.560	68.610	67.500	71.680	76.680
09:00:00	58.750	68.780	65.130	71.100	75.580
09:15:00	58.240	68.200	67.490	71.200	75.080
09:30:00	59.020	67.270	66.340	71.230	74.930

09:45:00	57.590	67.220	65.530	72.690	73.580
10:00:00	56.890	59.100	64.900	71.140	72.530
10:15:00	59.010	63.520	64.810	70.590	71.740
10:30:00	59.290	65.220	67.500	70.790	71.270
10:45:00	58.060	72.230	67.080	69.990	72.750
11:00:00	57.270	71.360	65.540	69.540	72.890
11:15:00	56.860	70.000	64.660	70.850	73.600
11:30:00	59.580	69.810	67.400	67.720	75.760
11:45:00	58.900	69.930	67.200	69.790	74.810
12:00:00	60.480	70.620	67.750	71.200	76.970
12:15:00	60.970	69.970	68.980	71.230	80.180
12:30:00	62.860	71.160	72.310	74.760	83.240
12:45:00	63.550	73.520	73.510	76.190	81.670
13:00:00	65.250	74.910	73.300	78.060	83.580
13:15:00	70.550	78.060	75.710	81.060	86.450
13:30:00	75.500	79.810	80.930	84.100	89.900
13:45:00	75.460	80.870	84.490	85.000	92.610
14:00:00	76.950	83.320	87.290	87.970	90.650
14:15:00	79.250	83.750	87.140	88.900	91.200
14:30:00	81.990	90.270	91.490	92.200	93.690
14:45:00	86.920	94.160	94.630	96.520	98.280
15:00:00	93.240	97.540	97.280	99.490	100.640

15:15:00	96.700	97.110	100.550	103.370	104.320
15:30:00	97.280	97.450	100.050	107.410	108.710
15:45:00	95.090	96.550	101.240	106.050	109.140
16:00:00	90.770	99.230	102.030	102.540	110.400
16:15:00	92.360	88.950	99.840	103.200	108.810
16:30:00	89.440	97.630	100.020	101.850	106.610
16:45:00	85.640	100.530	97.360	97.290	105.560
17:00:00	83.780	95.230	93.562	93.750	102.430
17:15:00	77.770	80.500	90.630	87.200	98.560
17:30:00	74.080	94.080	95.850	84.720	91.410
17:45:00	71.220	93.340	87.790	81.830	86.920
18:00:00	69.290	86.070	82.780	80.400	81.510
18:15:00	66.700	82.230	78.340	75.640	74.260
18:30:00	65.140	78.240	75.180	68.860	70.850
18:45:00	64.170	75.360	72.420	67.940	72.920
19:00:00	63.140	72.820	70.900	67.680	73.600
19:15:00	61.340	72.290	70.180	67.770	73.200
19:30:00	63.200	74.730	71.010	71.790	74.460
19:45:00	65.690	77.810	73.040	72.730	76.960
20:00:00	71.870	82.130	77.760	77.140	82.560
20:15:00	74.820	86.200	84.250	83.110	87.410
20:30:00	80.580	92.170	90.530	86.540	93.660

20:45:00	85.870	97.380	95.830	91.680	97.590
21:00:00	92.350	102.080	102.010	98.530	102.530
21:15:00	100.320	108.760	109.480	104.100	110.980
21:30:00	109.020	117.220	118.840	112.320	118.690
21:45:00	114.320	126.610	125.740	120.450	125.270
22:00:00	121.270	137.710	137.680	131.540	132.900
22:15:00	130.260	146.650	143.760	144.360	140.080
22:30:00	136.350	153.610	155.120	152.510	150.330
22:45:00	143.100	160.100	163.130	161.330	160.740
23:00:00	150.980	164.290	169.510	164.260	167.570
23:15:00	159.350	169.850	175.530	170.600	177.320
23:30:00	166.690	176.460	179.680	175.520	184.110
23:45:00	171.950	179.720	188.900	181.000	190.200

Table II: Normalized load data of TPDDL from 12 April 2012 – 15 April 2012

Time (in Hrs)	**Normalized load 11-04-2012**	**Normalized load 12-04-2012**	**Normalized load 13-04-2012**	**Normalized load 14-04-2012**	**Normalized load 15-04-2012**
00:00:00	0.7306	0.8139	0.8486	0.9000	0.8731
00:15:00	0.7229	0.8122	0.8476	0.8985	0.8783
00:30:00	0.7074	0.7980	0.8164	0.8762	0.8664
00:45:00	0.7027	0.7779	0.7893	0.8662	0.8511
01:00:00	0.7085	0.7695	0.7707	0.8404	0.8369
01:15:00	0.6959	0.7426	0.7506	0.8203	0.8283
01:30:00	0.6883	0.7284	0.7225	0.8055	0.8089
01:45:00	0.6653	0.7144	0.7278	0.7912	0.7767
02:00:00	0.6472	0.6998	0.7140	0.7734	0.7391
02:15:00	0.6295	0.6733	0.6961	0.7480	0.7427
02:30:00	0.6128	0.6540	0.6781	0.7302	0.7338
02:45:00	0.5993	0.6463	0.6523	0.7111	0.7163
03:00:00	0.5857	0.6278	0.6404	0.6968	0.6961
03:15:00	0.5618	0.5984	0.6178	0.6868	0.6715
03:30:00	0.5474	0.5872	0.6097	0.6671	0.6544
03:45:00	0.5273	0.5755	0.5970	0.6507	0.6522
04:00:00	0.5040	0.5506	0.5732	0.6240	0.6474
04:15:00	0.4763	0.5237	0.5486	0.5907	0.6282

04:30:00	0.4451	0.4983	0.5224	0.5713	0.6081
04:45:00	0.4295	0.4817	0.5004	0.5452	0.5862
05:00:00	0.4132	0.4606	0.4831	0.5256	0.5583
05:15:00	0.3878	0.4358	0.4479	0.4730	0.5334
05:30:00	0.3410	0.3944	0.4118	0.4458	0.4940
05:45:00	0.3006	0.3534	0.3774	0.3944	0.4510
06:00:00	0.2653	0.3181	0.3357	0.3661	0.4096
06:15:00	0.2395	0.2884	0.3043	0.3480	0.3716
06:30:00	0.2115	0.2687	0.2738	0.3303	0.3461
06:45:00	0.2002	0.2564	0.2581	0.3140	0.3214
07:00:00	0.1908	0.2493	0.2460	0.3017	0.3023
07:15:00	0.1781	0.2167	0.2206	0.2740	0.2796
07:30:00	0.1682	0.2139	0.2038	0.2615	0.2589
07:45:00	0.1642	0.1959	0.2013	0.2447	0.2511
08:00:00	0.1593	0.1876	0.1879	0.2261	0.2435
08:15:00	0.1403	0.1844	0.1774	0.1987	0.2221
08:30:00	0.1320	0.1814	0.1693	0.1872	0.2145
08:45:00	0.1159	0.1698	0.1632	0.1881	0.2179
09:00:00	0.1111	0.1708	0.1491	0.1847	0.2113
09:15:00	0.1080	0.1674	0.1632	0.1853	0.2084
09:30:00	0.1127	0.1618	0.1563	0.1854	0.2075
09:45:00	0.1042	0.1615	0.1515	0.1941	0.1994

10:00:00	0.1000	0.1132	0.1477	0.1849	0.1932
10:15:00	0.1126	0.1395	0.1472	0.1816	0.1885
10:30:00	0.1143	0.1496	0.1632	0.1828	0.1857
10:45:00	0.1070	0.1914	0.1607	0.1780	0.1945
11:00:00	0.1023	0.1862	0.1515	0.1754	0.1953
11:15:00	0.0998	0.1781	0.1463	0.1832	0.1996
11:30:00	0.1160	0.1770	0.1626	0.1645	0.2124
11:45:00	0.1120	0.1777	0.1614	0.1769	0.2068
12:00:00	0.1214	0.1818	0.1647	0.1853	0.2196
12:15:00	0.1243	0.1779	0.1720	0.1854	0.2388
12:30:00	0.1356	0.1850	0.1919	0.2065	0.2570
12:45:00	0.1397	0.1991	0.1990	0.2150	0.2476
13:00:00	0.1498	0.2074	0.1978	0.2261	0.2590
13:15:00	0.1814	0.2261	0.2121	0.2440	0.2761
13:30:00	0.2109	0.2366	0.2432	0.2621	0.2967
13:45:00	0.2106	0.2429	0.2644	0.2675	0.3128
14:00:00	0.2195	0.2575	0.2811	0.2852	0.3011
14:15:00	0.2332	0.2600	0.2802	0.2907	0.3044
14:30:00	0.2495	0.2989	0.3061	0.3104	0.3192
14:45:00	0.2789	0.3220	0.3248	0.3361	0.3466
15:00:00	0.3166	0.3422	0.3406	0.3538	0.3606
15:15:00	0.3372	0.3396	0.3601	0.3769	0.3826

15:30:00	0.3406	0.3416	0.3571	0.4010	0.4087
15:45:00	0.3276	0.3363	0.3642	0.3929	0.4113
16:00:00	0.3018	0.3522	0.3689	0.3720	0.4188
16:15:00	0.3113	0.2910	0.3559	0.3759	0.4093
16:30:00	0.2939	0.3427	0.3570	0.3679	0.3962
16:45:00	0.2713	0.3600	0.3411	0.3407	0.3900
17:00:00	0.2602	0.3284	0.3185	0.3196	0.3713
17:15:00	0.2244	0.2407	0.3010	0.2806	0.3483
17:30:00	0.2024	0.3216	0.3321	0.2658	0.3057
17:45:00	0.1854	0.3172	0.2841	0.2486	0.2789
18:00:00	0.1739	0.2738	0.2542	0.2401	0.2467
18:15:00	0.1584	0.2510	0.2278	0.2117	0.2035
18:30:00	0.1492	0.2272	0.2090	0.1713	0.1832
18:45:00	0.1434	0.2100	0.1925	0.1658	0.1955
19:00:00	0.1372	0.1949	0.1835	0.1643	0.1996
19:15:00	0.1265	0.1917	0.1792	0.1648	0.1972
19:30:00	0.1376	0.2063	0.1841	0.1888	0.2047
19:45:00	0.1524	0.2246	0.1962	0.1944	0.2196
20:00:00	0.1892	0.2504	0.2243	0.2206	0.2529
20:15:00	0.2068	0.2746	0.2630	0.2562	0.2818
20:30:00	0.2411	0.3102	0.3004	0.2766	0.3191
20:45:00	0.2727	0.3412	0.3320	0.3073	0.3425

21:00:00	0.3113	0.3692	0.3688	0.3481	0.3719
21:15:00	0.3587	0.4090	0.4133	0.3813	0.4223
21:30:00	0.4106	0.4594	0.4691	0.4302	0.4682
21:45:00	0.4422	0.5154	0.5102	0.4787	0.5074
22:00:00	0.4836	0.5815	0.5813	0.5447	0.5528
22:15:00	0.5371	0.6348	0.6175	0.6211	0.5956
22:30:00	0.5734	0.6762	0.6852	0.6697	0.6567
22:45:00	0.6136	0.7149	0.7329	0.7222	0.7187
23:00:00	0.6606	0.7399	0.7710	0.7397	0.7594
23:15:00	0.7104	0.7730	0.8068	0.7775	0.8175
23:30:00	0.7542	0.8124	0.8315	0.8068	0.8579
23:45:00	0.7855	0.8318	0.8865	0.8394	0.8942

REFERENCES

[1] I.J.Nagrath and D.P. Kothari, Modern Power System Analysis, Tata McGraw Hill Publishers, 2003.

[2] Eugene A. Fienberg and Dora Genethliou, State University of New York, Stony Brook, “Chapter 12-Load Forecasting”, pp.270-285

[3] Article in fuzzy logic: Wikis “http://www.thefullwiki.org/fuzzy_logic”

[4] R.J.Campbell and K. Adamson, “Methodologies For Load Forecasting”, 3rd International IEEE Conference on Intelligent Systems, pp. 800-806, Sep. 2006

[5] J. W. Taylor, “An Evaluation of Methods For Very Short-Term Load Forecasting Using Minute by Minute British data,” Int. J. Forecast., vol. 24, pp. 645–658, 2008

[6] http://mathworks.com

[7] M. Rizwan, Dinesh Kumar, Rajesh Kumar, “Fuzzy Logic Approach for Short Term Electrical Load Forecasting” Electrical and Power Engineering Frontier, Dec. 2012, Vol. 1 Iss. 1, PP. 8-12

[8] S. C. Pandian, K. Duraiswamy, C. C. A. Rajan, N. Kanagaraj, “Fuzzy Approach for Short Term Load Forecasting”, Electrical Power System Research, vol. 76, pp. 541-548, 2006

[9] Forecasting load data source, “TATA Power Delhi Distribution Limited, Pitampura”

[10] Article in: Wikis http://www.thefullwiki.org/neural_network

[11] John Yen and Reza Langari, “Artificial Intelligence, Control and Information ,”Pearson education ,first Indian reprint,2003 pp 21-53,60-78,109-127

[12] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural Networks for Short-Term Load Forecasting: A Review and Evaluation”, IEEE Transaction on Power Systems, vol. 16, no. 1, pp. 44–55, 2001.

[13] S.Sapna , Dr. A. Tamilarasi and M. Pravin Kumar,” Backpropagation Learning Algorithm Based on Levenberg Marquardt Algorithm”

[14] LipikaSharma, Monika Chakravarti, Abhijeet Dutta and N. Adhikari, ”Neural Network Based Approach for Short-Term Load Forecasting”, International Journal of Science, Environment and Technology, vol.1, no .5, pp.416-424, 2012

[15] K.Y.Lee, J. H. Park,‘Short-Term Load Forecasting Using an Artificial Neural Network”, Transactions on Power Systems, Vol. 7. No. 1, February 1992

[16] Ireri T. Gichovi, Daniel K. Murage and Nicodemus Abungu,” Short Term Load Forecasting Using Artificial Neural Networks”

[17] F.J. Marin, F. Garcia-Lagos, G.Joya and F.Sandoval,” Global Model for Short Term Load Forecasting Using Artificial Neural Networks”

[18] Efstathios Paparoditis, Theofanis Sapatinas,” Short-Term Load Forecasting: The Similar Shape Functional Time-Series Predictor”, IEEE Trans. on Power Syst, Vol. 28, NO. 4, NOVEMBER 2013

[19] A.J. Al-Shareef, E.A. Mohamed and E.Al-Judaibi, King Abdulaziz University,Qassim University and 2 SEC-WOA, Saudi Arabia,’ Next 24-Hours Load Forecasting Using Artificial Neural Network (ANN) for the Western Area of Saudi Arabia’, JKAU: Eng. Sci., Vol.19 No.2, pp: 25-40 (2008A.D. / 1429 A.H.)

[20] G. Strang and T. Nguyen, Wavelets and Filter Banks, 2nd ed. Wellesley, MA: Wellesley Cambridge Press, 1997.

[21] Y. Chen,P.B.Luh,C.Guan, Y. G. Zhao,L.D.Michel,M.A. Coolbeth, P. B. Friedland, and S. J. Rourke, “Short-Term Load Forecasting: Similar Day-Based Wavelet Neural Networks,” IEEE Trans. Power Syst., vol. 25, no. 1, pp. 322–330, Feb. 2010

[22] Che Guan, Peter B. Luh, Laurent D. Michel, Yuting Wang, and Peter B. Friedland, Member, IEEE Very Short-Term Load Forecasting: Wavelet Neural Networks With Data Pre-Filtering", IEEE Trans. Power Syst., VOL. 28, NO. 1, FEBRUARY 2013

[23] C. Guan, P. B. Luh, M. A. Coolbeth, Y. Zhao, L. D. Michel, Y. Chen, C. J. Manville, P. B. Friedland, and S. J. Rourke, "Very Short-Term Load Forecasting: Multilevel Wavelet Neural Networks with Data Pre-Filtering," in *Proc. IEEE Power and Energy Society 2009 General Meeting*, Calgary, AB, Canada, Jul. 2009

[24] Data pre-filtering data source,"www.bpa.gov.in".