WIND SPEED ESTIMATION USING ARTIFICIAL NEURAL NETWORKS

By

NS Zeeshan Ali Cheema

00000172288

MS-ME-16

Submitted to the Department of Mechanical Engineering in Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

In

MECHANICAL ENGINEERING

Thesis Supervisor

Dr. Imran Shafi, PhD

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2020

WIND SPEED ESTIMATION USING ARTIFICIAL NEURAL NETWORKS

By

NS Zeeshan Ali Cheema

00000172288

MS-ME-16

A thesis submitted in fulfillment of partial requirements for the Degree

MASTER OF SCIENCE

In

MECHANICAL ENGINEERING

Thesis Supervisor

Dr. Imran Shafi, PhD

Thesis Supervisor’s Signature:____

Department of Mechanical Engineering

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2020

Copyright Statement

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

In the name of Allah, the most Beneficent and the most Merciful

Declaration

I hereby affirm that this thesis Title **"WIND SPEED ESTIMATION USING ARTIFICIAL NEURAL NETWORKS"** is absolutely based upon my own personal hard work under the valuable guidance of my supervisor Dr Imran Shafi. The Contents have not been plagiarized and sources used are cited. No part of work presented in thesis has been submitted in favor of any application of other degree of qualification to this or any other university or institute of learning.

………………………

Zeeshan Ali Cheema

LANGUAGE CORRECTNESS CERTIFICATE

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

NS Zeeshan Ali Cheema

00000172288

Signature of Supervisor

(Dr. Imran Shafi)

ACKNOWLEDGEMENT

First and foremost, I would like to thank my Almighty Allah for all His countless blessings and benevolence which embraced me to complete this incredible task in terms of integrity and completeness.

Secondly, there is long list of people without whom I would have not been able to complete this assignment successfully. I would like to communicate special recognition and gratitude to my advisor,

Dr. Imran Shafi for his extended support throughout my thesis tenure. His untiring efforts and advice helped me to move around every predicament during my research work.

I would also like to thank Assistant Professor **Dr. Sajidullah Butt** and **Dr. Naveed Akmal Din**, for serving as my Guidance and Evaluation committee even at hardship. I obliged for letting my defense as enjoyable moments and for brilliant comments and suggestions.

Special thanks to my family for all the sacrifices that they made and who were always in my support in dire moments. Conclusively, I would like to thank the entire honorable faculty of Department of Mechanical Engineering, whose professional approach and vision groomed me as a sound person both technically and morally.

Zeeshan Ali Cheema, 2016

DEDICATION

- I dedicate my thesis to my family and friends. A special feeling of gratitude to my loving parents, whose words of encouragement were always there in my moral support.
- I would place sincere thanks to Dr. Imran Shafi for his guidance and support throughout this study and specially his confidence in me.
- I pray ALLAH, for always being kind by way His countless blessings.

ABSTRACT

In this masters thesis, two type of Artificial Neural Networks, Feed Forward Back Propagation Neural Network (FFBP NN) and Nonlinear Autoregressive Neural Network (NAR NN) are used for wind speed estimations and predictions. Feed Forward Back Propagation Neural Network is used to estimate wind speed based on three meteorological parameters namely, Temperature, Pressure and Humidity. While Nonlinear Autoregressive Neural Network is used to predict upcoming wind speeds without any input parameters. Feed Forward Back Propagation Neural Network with Levenberg training algorithm needs input and target parameters for its training. In this study an additional input parameter known as humidity is considered and results are compared with existing study. Input parameters that are used for training are Temperature, Pressure and Humidity while target values are of wind speed. Networks output value is compared with real time data of wind speed values for computation of performance parameter. 70% of data is used as training data and remaining 30% data is used for validation and testing. In case of Nonlinear Autoregressive Neural Network, the network is trained by taking previous wind speed as input and data point next to those inputs as target. The number of previous data points that are taken as input are dependent on the set delay value.

Both networks are trained with real data and results showed an improvement in accuracy due to consideration of additional input parameter. This study used multilayered Artificial Neural Network for medium to long term wind speed predictions. Daily, monthly and yearly wind speed predictions are part of this study. Weibull analysis that is most common and popular wind speed estimation approach is also discussed in this study. This study contains a brief review of Weibull curve analysis and its limitations. Also the improvement in results due to consideration of an additional parameter in comparison with an existing study is discussed. Matlab is used for building Neural Architecture and running algorithms.

Table of Contents

List of Figures

List of tables

LIST OF ACRONYMS

PDF	Probability Density Function
ANN	Artificial Neural Networks
FFBP NN	Feed forward Back Propagation Neural Network
NAR Net	Nonlinear Autoregressive Network
LM	Levenberg Marquardt
GDM	Gradient Descent with Momentum
Mu	Momentum
Mu_inc	Momentum Increase
SSE	Sum Squared Error
MSE	Mean Square Error
MAE	Mean Absolute Erro
RMSE	Root Mean Square Error

Chapter 1: INTRODUCTION

Wind speed estimation plays an important role in different applications and because of this reason it is of great concern for mankind. The major application areas are weather forecasting, aviation and maritime operations, design of high rise structures and site selection for wind turbines. Renewable energy gained popularity soon with the increase in demand of energy and environmental concerns. The fossil fuels increased the environmental pollution, as a result the renewable energy resources attracted the attention of experts throughout the world. To fulfil this huge demand of energy that is increasing day by day with population, sustainable energy resources are long term and environment friendly. These resources got high value of attention in many countries all over the world. Wind power generation emerged as most popular and mature renewable energy generation technology which worked effectively with other renewable technologies in hybrid form as well. For the applications in wind energy generation prediction of wind speed has great importance.

Wind speed depends on different parameters. These parameters have major or minor effects on wind speeds depending on the location where wind speed is measured. Wind is basically due to the motion of air molecules and these air molecules move only when a difference of pressure appears. Air particles move from a point where there is high pressure towards a point of low pressure. Hence pressure gradient is a important factor that effects wind speed.

Macroscopically wind speed is effected by pressure gradient, weather of the region and Rossby waves. Rossby waves are inertial waves that are caused due to motion of fluids. Here one thing is important to note that these ways do not effect wind speed near the surface of earth. At point closer to earth, wind speeds are greatly effected by different other parameters like hills, construction and traffic etc.

In lot of modern studies, Weibull distribution has been considered a most appropriate approach to annual mean speed variation. It has ability to represent various distributed characteristics after once its parameters are calculated. The parameters of Weibull are shape and scale parameter. These parameters have significant importance in case of two parameter Weibull distribution function. While in case of three parameter Weibull distribution function, there is addition of an additional parameter known as location parameter. Weibull distribution function after estimating these above mentioned parameters can be used to model wind speed changes and estimation of future wind speeds.

1.1 AIM OF STUDY

The conventional approach of modeling was based on to fit probability density function to a given probability density function model to project factors having statistical importance, like mean and variance. However these models do not have variational properties based on time, in addition with excluding the cross dependence between other meteorological data.

Artificial Neural networks covered this weakness and addressed the problem of time-based variational trends in wind speed. Also ANN are previously used for prediction of different parameters, like Kuo, P [1] predicted electricity prices, De Filippo [2] used it for electricity price optimization, Cincotti, S [3] used it for forecasting of electricity spot price and Aliihsan [12] for land teemperature. In this study Feed Forward Backpropagation Neural Network is used for input based wind speed estimations while Nonlinear Autoregressive Neural Network is used to prediction of wind speed in future with appreciable accuracy.

Also this study shows how results of Neural networks can be improved by considering appropriate input parameters.

1.2 FEATURES OF THE CITY WHERE DATA COLLECTED

The data was collected in Istanbul, situated in north-western Turkey in Marmara Region. Turkey occupies a unique position of existing partly in Asia and Europe. Its total area is of 785,347 square kilometers. Bosphorus connects sea of Marmara to Black sea and divides city into European, Thracian and Asian, Anatolian sides. Thracian comprises of historic and economic centers. Further the city is divided by golden horns which is its natural harbor.

According to the model of Rome, peninsula is historically characterized by seven hills that are overstepped by imperial mosques. Topkapi Palace is situated on the easternmost of these hills. On the opposite side Golden Horn there is another conical hill present in modern Beyoglu district. Topography showed that Beyoglu were constructed by the help of terraced retaining walls and the road were in form of steps. Uskudar situated on the Asian side has hilly features. The highest point in Istanbul is camlica hill having altitude 288 meters, also the northern half of Istanbul has higher elevation as compared to south coast having steep cliffs. Northern end of Istanbul opens into Black sea.

1.2.1 CLIMATE

Istanbul has a unique borderline weather, it has Mediterranean, humid subtropical and ocean nature climate. The major reason behind this type of weather is its location that is in transitional climatic zone. Precipitation rate in summer ranges from 20-65 mm depending on the conditions. The city itself cannot be categorized as solely Mediterranean subtropical because of its size and diversity in topography and maritime location. Most important factor effecting the weather of the city is presence of two bodies of water in north and south sides. The north half of city expresses the characteristics of ocean and humid subtropical climate nature because of the presence of black sea. The populated area of city is dry, warmer and less effected by humidity which is actually the south to the city on sea of Marmara. The annual precipitation on northern region is twice as much as on southern region. There is a remarkable annual mean temperature difference on north and south coasts. Areas of province that are away from both seas have night-day and winter-summer temperature differences, especially in winters some parts have below freezing temperatures.

Istanbul's consistently high rate of humidity reaches 80% mostly in mornings. Because of this reason fog is very common in region. This is common in autumns and winters. During the summer the fog and humid conditions tend to dissipate at midday but lingering humidity exacerbates high temperatures. During summers the high temperature rises to 29 °C with rare rain fall and has only fifteen days of measurable precipitation. Thunderstorms are also very common in summers.

Winter has remarkable low temperatures as compared to other cities that are between 1-4°C. Snow from lake effect caused by Black sea is common and difficult to fore cast. Fog and city infrastructure makes it challenging to predict wind speed. Sometimes there are large fluctuations in temperature. Istanbul has annually 130 days of significant precipitation up to 180mm. The highest and lowest temperature ranges recorded in city center at Marmara coast are 40 to -16 °C. The maximum recorded rainfall and snow in one day are 227 mm and 80 cm respectively.

1.3 ORGANIZATION OF THE THESIS

The present thesis comprises of six chapters.

Chapter 1 starts with a discussion on the general introduction of methods used for wind estimations. Also, it contained some information related to the region where wind speed data was collected. There is also a brief intro duction of aim of study and goals.

Chapter 2 Presents a literature review of the published articles and work of different researchers in field of wind speed predictions. It also contains the step wise achievements of different researchers by using different approaches.

Chapter 3 Presents a modern and commonly used method for wind speed estimation, that is Weibull curve analysis.

Chapter 4 Presents introduction, design and working philosophy of Artificial Neural Networks.

Chapter 5 Presents the test of trained Artificial neural network for wind speed estimations and predictions.

Chapter 6 Presents conclusions and future scope of work.

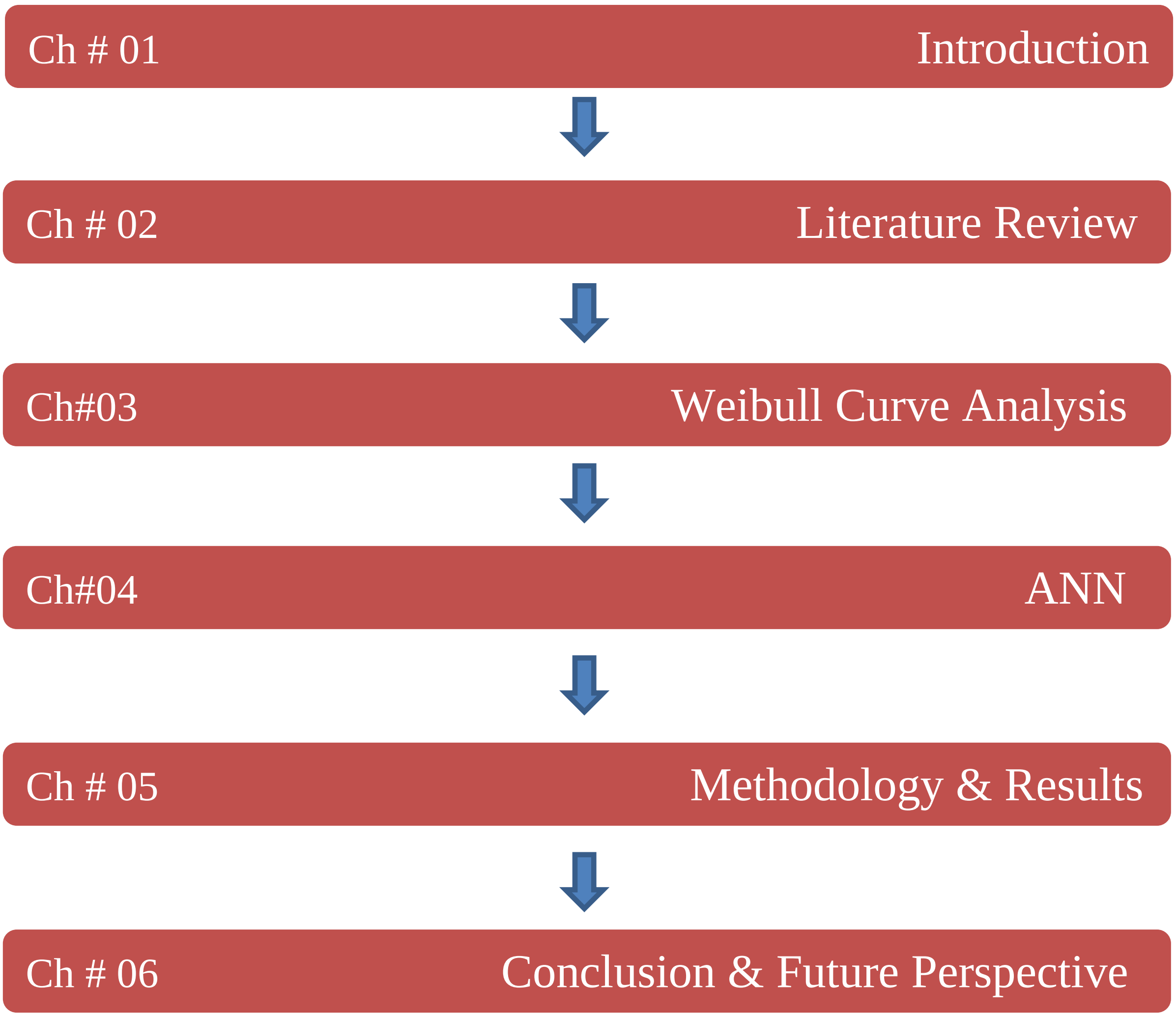

Figure 1.1 Deposition of Thesis Repor

A list of references and appendices are included at the end

CHAPTER 2: LITERATURE REVIEW

This chapter contains a brief review of research publications and literature from books to make reader clear about the methodologies that used for wind speed estimations. It also contains how researchers tried to improve accuracy in predictions. Most commonly used and modern method for wind speed predictions is discussed in detail in next chapter.

2.1 ARTIFICIAL NEURAL NETWORKS

In time series problems [1-2], Cincotti-et-al used three different methods to model the issue of discrete time univariate econometric model and two artificial techniques. The methodology of support vector machine learning shows better forecasting for price time series [3]. Short term wind speed prediction is of great importance for farm power output. This enables the power delivery system to adjust the delivery schedule. Effective prediction also improves the wind power penetration rate and reduce the impact of wind power on the power grid.

If the wind speed pattern is determined by any approach even then there has inherent regular trends which provides feasibility for wind speed forecasting. Actual practice shows mostly that wind speed fluctuates randomly and unstable. Wind speed varies time to time and with altitude at same place. There are many parameters that effect wind speed like seasonality, temperature, humidity, air pressure and other parameters that are due to location on map or other with other dominant features like sea, mountains and deserts etc.

There are many approaches for wind speed forecasting. According to [4] Bayesian model for structural break down proposed to conduct exact short term wind speed forecast. He used actual experimental data and computed mean square error (MSE), mean absolute error (MAE) and root mean square error (RMSE). This study can be effectively applied to for wind power prediction and wind power planning. The precision of this method is very high but can be used for ultrashort wind speed prediction for few minutes to hours. Study [5] proposed wind speed testing combined with solar radiations based on extreme learning machines (ELM) and principal components analysis (PCA). According to author PCA can be used to reduce the data dimensions with reduction in approach complexity in training. Results included that train by ELM is much faster compared to perceptron repeated layers, radial based networks and least squares support vector machines.

Models that involve statistical series of time possibly predict the future values of wind speed based on the distribution of past wind speed data values. This is done with the help of special parameters obtained from distribution. Fortuna et [6] time series data can be characterized by a data type that is time varying. Majorly time series clustering approaches are characterized into three categories depending on the fact whether they work directly with raw form of data, indirectly with features of raw data that are extracted from raw data or indirectly with the built models based on raw data. The most significant assumption in building time series model is stationary. Stationary time series can be conveniently regenerated by using its standard deviation and mean.

CHAPTER 3: WEIBULL CURVE

This chapter describe how Weibull curve works in detail. Weibull is most popular approach in life data estimations. It is used to find out failure rates and checking the reliability of products. Due to its versatile nature, it can be used for wind speed estimations.

3.1 OVERVIEW OF WEIBULL CURVE ANALYSIS

Weibull data analysis is also known as life data analysis. The researchers working on it make predictions about the life of all products by fitting data in statistical distribution. Then the distribution based on parameters is used for the predictions of reliability or failure probability at specific instance. The mean life and failure rate are also the quantities that can be known by this approach. The requirements for the failure data analysis is;

- Gathering data for the specimen
- Selection of distribution that fit the data.
- Estimation of parameters that fit the data distribution.
- Generation of plots that estimate the life characteristics.

This study explains the basic concept and use of Weibull analysis approach for wind speed estimation, that is commonly used by different researchers. Matlab is used for analyzing data for Weibull analysis.

3.1.1 THREE PARAMETER WEIBULL PDF

Equation below shows the three parameter Weibull probability densityfunction. The Weibull distribution model can be applied in various forms, including Weibull one parameter, two parameters, three parameters or mixed form. Piotr Wais [16] and Jos´e A [18] describes two and three parameter Weibull in detail. It is the duty of data analyst to choose the life distribution that is more appropriate to the data by experience and goodness of fit test.

$$\mathrm{f}(t) = \frac{\beta}{\eta}\left(\frac{\mathrm{t}-\gamma}{\eta}\right)^{\beta-1} \mathrm{e}^{-\left(\frac{\mathrm{t}-\gamma}{\eta}\right)^{\beta}} \tag{1}$$

where:

$\mathrm{f(t)} \geq 0$ and $\mathrm{t} \leq \gamma$

$\beta > 0$

$\eta > 0$

$-\infty < \gamma < +\infty$

3.1.2 TWO PAMETER WEIBULL PDF

The two parameter Weibull probability density function is obtained from three parameter equation by putting ɣ=0. It is given as;

$$\mathrm{f}(t) = \frac{\beta}{\eta}\left(\frac{\mathrm{t}}{\eta}\right)^{\beta-1} \mathrm{e}^{-\left(\frac{\mathrm{t}}{\eta}\right)^{\beta}} \tag{2}$$

3.1.3 ONE PARAMETER WEIBULL PDF

This type of probability function is obtained by setting ɣ = 0 and considering β = C = constant. Weibull one parameter probability function is given as;

$$\mathrm{f}(t) = \frac{C}{\eta}\left(\frac{\mathrm{t}}{\eta}\right)^{C-1} \mathrm{e}^{-\left(\frac{\mathrm{t}}{\eta}\right)^{C}} \tag{3}$$

in one parameter Weibull Probability density function has only one unknown term, that is scale parameter η. Main advantage of this distribution function is, we can have knowledge of few or no failures. The value of shape factor is known from the past experience of similar specimen.

3.1.4 MEAN OF FUNCTION

The mean of Weibull probability density function is given by;

$$\bar{T} = \gamma + \eta.\Gamma\left(\frac{1}{\beta} + 1\right) \tag{4}$$

Where $\Gamma\left(\frac{1}{\beta} + 1\right)$ is the gamma function that can be evaluated by;

$$\Gamma(\eta) = \int_{0}^{\infty} e^{-x}x^{n-1}\,dx \tag{5}$$

For the two parameter case mean can reduce to;

$$\bar{T} = \eta.\Gamma\left(\frac{1}{\beta} + 1\right)$$

3.1.6 MODE OF WEIBULL DISTRIBUTION

The mode of Weibull distribution is given by;

$$\breve{T} = \gamma + \eta\left(1 - \frac{1}{\beta}\right)^{1/\beta} \tag{6}$$

3.1.7 STANDARD DEVIATION

The standard deviation of Weibull curve is given as;

$$\sigma_T = \eta.\sqrt{\Gamma\left(\frac{2}{\beta} + 1\right) - \Gamma\left(\frac{1}{\beta} + 1\right)^2} \tag{7}$$

3.1.8 WEIBULL RELIABILITY FUNCTION

The three parameter Weibull cumulative density function can be expressed as;

$$\mathrm{F}(t) = 1 - e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}} \tag{8}$$

This is also known as unreliability of specimen.

3.2 LIFE DATA

Life data is a term used for measurements showing product life. This data can be in units of time, distance, cycles or any other that shows the successful operation of product. The most common measurement is in time and is of different types as it provides different information about equipment life. The method used for analysis also varies according to data type. The complete data has exact time of failure, while in case of suspended data the product operated successfully. It is also known as right censored data. In case of interval or left censored data, the exact time of specimen's failure is unkown but it fails in specific time range.

3.3 LIFE TIME DISTRIBUTIONS

The distributions are formulated statistically by mathematicians, statisticians and engineers for analyzing data. Data analysts use these mathematical models for representation of different behaviors. One of the most important mathematical functions that defines distribution is "probability density function''. Probability distribution for failures in different structures are also found by Weibull analysis, as discussed by Yu-Cai Zhang, Hui-Qin Zhao and Wenchun Jiang in their study [7] and fitting of data in three parameter Weibull distribution is discussed by Asghar Moeini [19]. Figure 3.1 shows Weibull distribution of sample data set of carsmalls in Matlab.

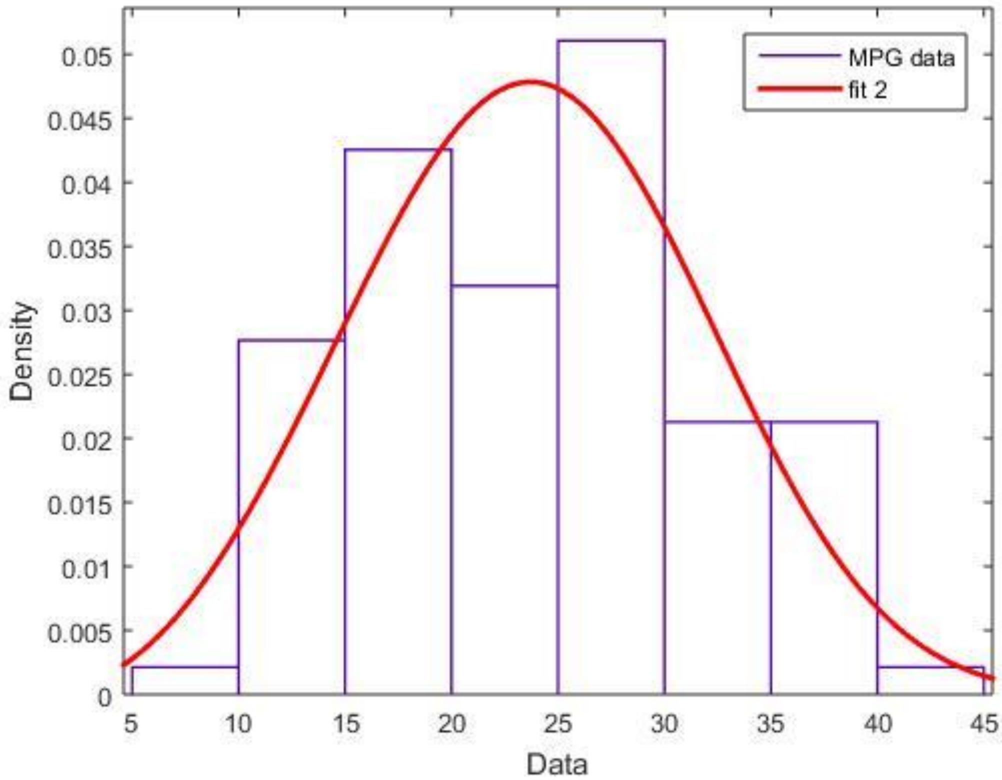

Figure 3.2 Weibull curve

3.4 PARAMETER ESTIMATION

In order to use the probability density function for life data analysis, it is required to estimate the values of parameters. Data analysts use the values of parameters that make the function of statistical model, most closely fit the life data. These parameters include shape, scale and location parameters of probability density function. The above equation shows thee three parameter probability density function. The scale parameter "ղ'' tells, where the maximum distribution of data lie. The shape parameter "β" defines the shape of the curve, which is basically the slope of gradient and location parameter "γ" defines the time based location of distribution.

3.5 CHARACTERISTICS OF WEIBULL DISTRIBUTION

This approach is widely used in reliability analysis of products due to its versatility. With the selection appropriate Weibull parameters it is used to model variety of problems. The effect of changing these parameters on curve is shown below.

3.5.1 SHAPE PARAMETER

The Weibull shape parameter β is also known as slope because it is equal to the slope of regressed line in plot. Different values of shape parameter have considerable effect on the behavior of curve and this make it versatile to different distributions. By putting β =1 we can reduce three parameter function to two parameters.

$$f(t) = \frac{1}{\eta} e^{-\frac{t-\gamma}{\eta}} \tag{8}$$

Where $1/\eta = \lambda$, known as failure rate. Shape factor is a pure number and is dimensionless. The effect of different values of shape factor on Weibull probability density function and cumulative density function is discussed by Chin-Diew Lai [8] as shown in figure;

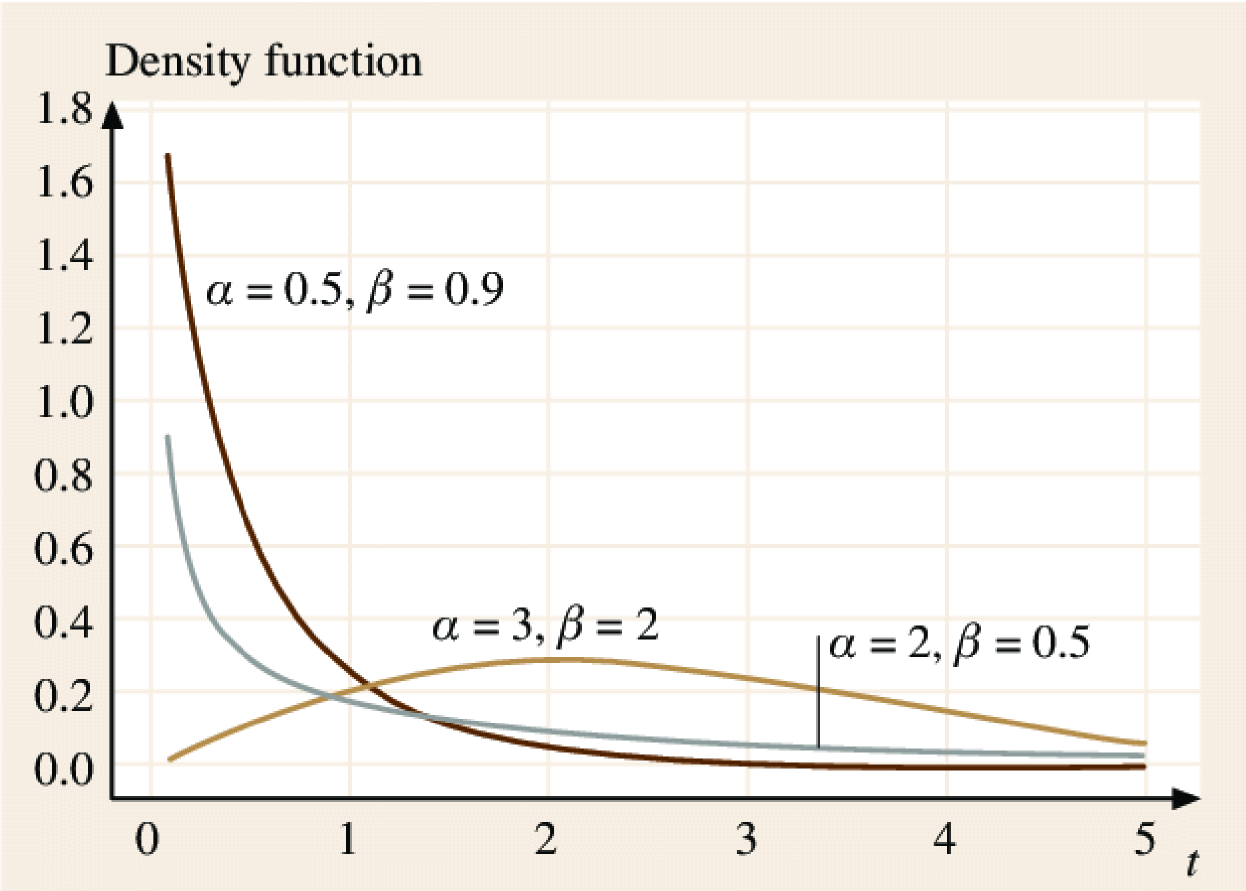

Figure 3.2 Weibull PDF for different β values [8]

From above figure it can easily be seen that increase in shape function leads to the right shifting of curve. If value of shape function is equal to 1, it becomes Rayleigh distribution.

3.5.1.1 EFFECT OF SHAPE FACTOR ON UNRELIABILITY

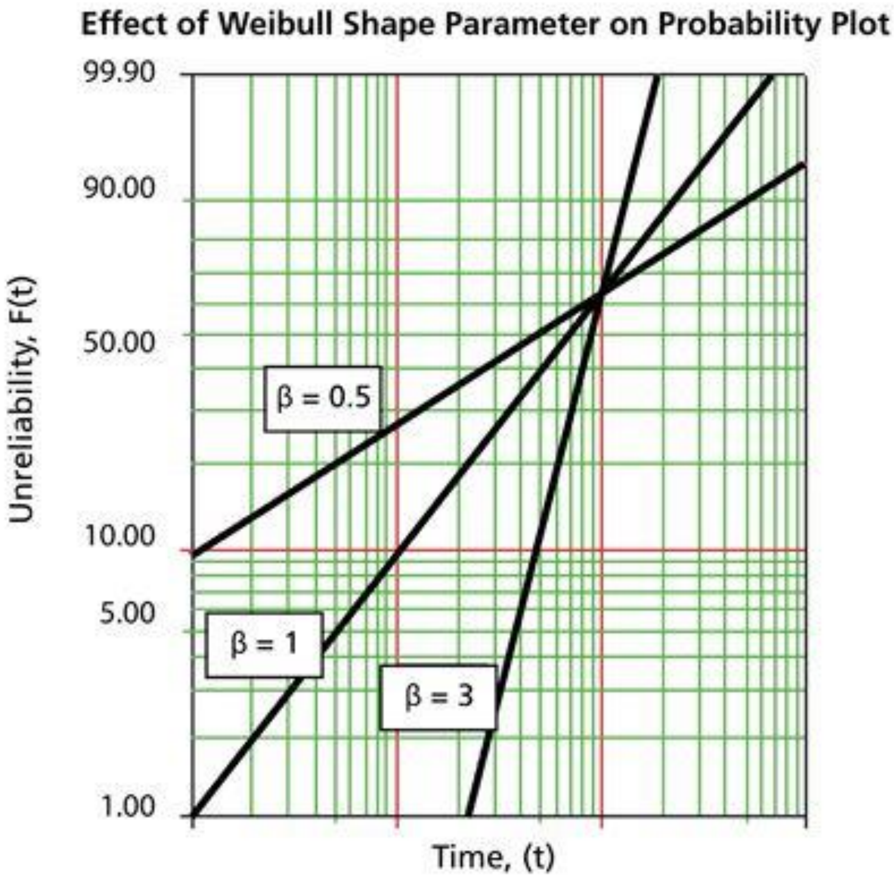

Figure 4.3 Effect of changing β on Unreliability [20]

The above figure clearly demonstrates why shape parameter is known as slope. All the three lines shown in above graph are at same value of η.

The effect of changing shape factor on reliability plot is shown below;

3.5.1.2 EFFECT OF SHAPE FACTOR ON RELIABILITY PLOT

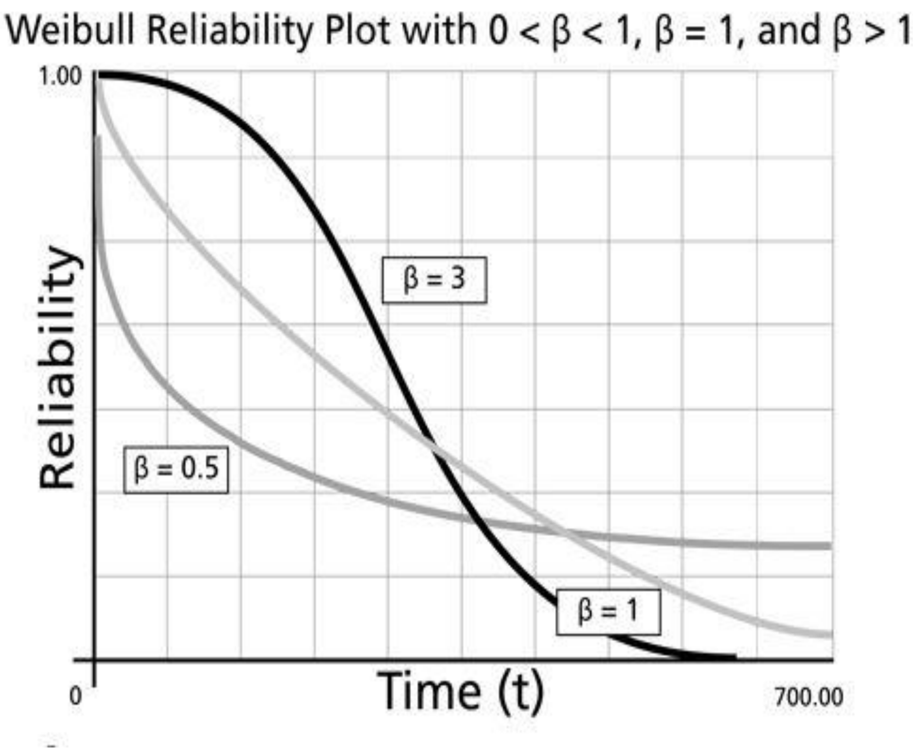

Figure 5.4 Effect of different β values on Reliability [20]

- R(t) decreases sharply for 0 < β < 1.
- R(t) decreases less sharply for 0 < β < 1 in convex shape.
- R(t) decreases as with similarly as increases for β > 1.

3.5.1.3 THE EFFECT OF SHAPE FACTOR ON FAILURE

The figure below shows the effect of Weibull shape parameter on failure rate of product. A comparison between the graphs drawn at values of shape factor greater and less than 1 are shown in figure 3.5 a:

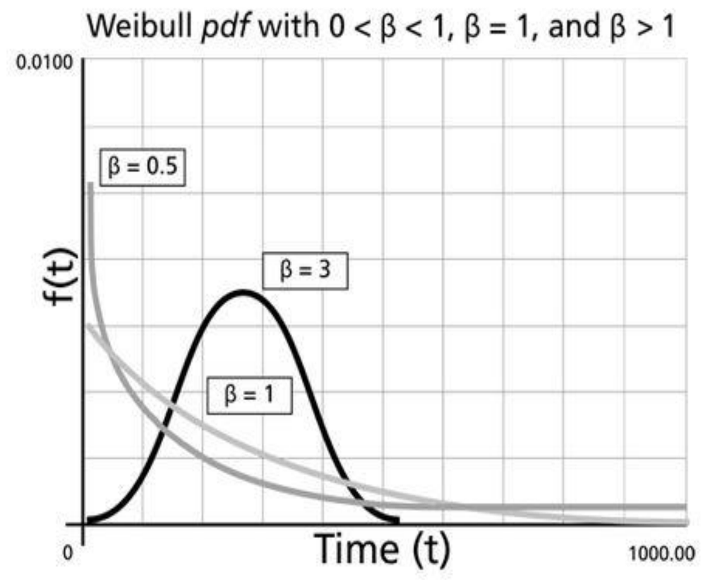

Figure 6.5a Effect of different β values on failure rate [20]

As shown in above figure $\beta < 1$ has a decreasing failure rate with time, $\beta = 1$ has constant failure rate that is consistent with failure rate and $\beta > 1$ have failure rates that increase with time. For $0 < \beta < 1$ failure rate decreases and is convex in shape, this makes it suitable for early type failures in specimen and these failures decreases with age. In case of manufactured product it points towards the problem in manufacturing process. $1 < \beta < 2$ the failure rate increases with less slope as t increases. At $\beta = 2$ Weibull distribution function attains the form of Rayleigh distribution. As if $\beta > 2$ the curve becomes convex and slope increases with increase in time. This shows the wear out life.

Cumulative Density Function (CDF) for strength analysis is used for computation of failure probability for different materials of the applied stress state. It allows prediction of percentage of failure as described by Trevor G. Aguirre [9] and shown in figure 3.5b.

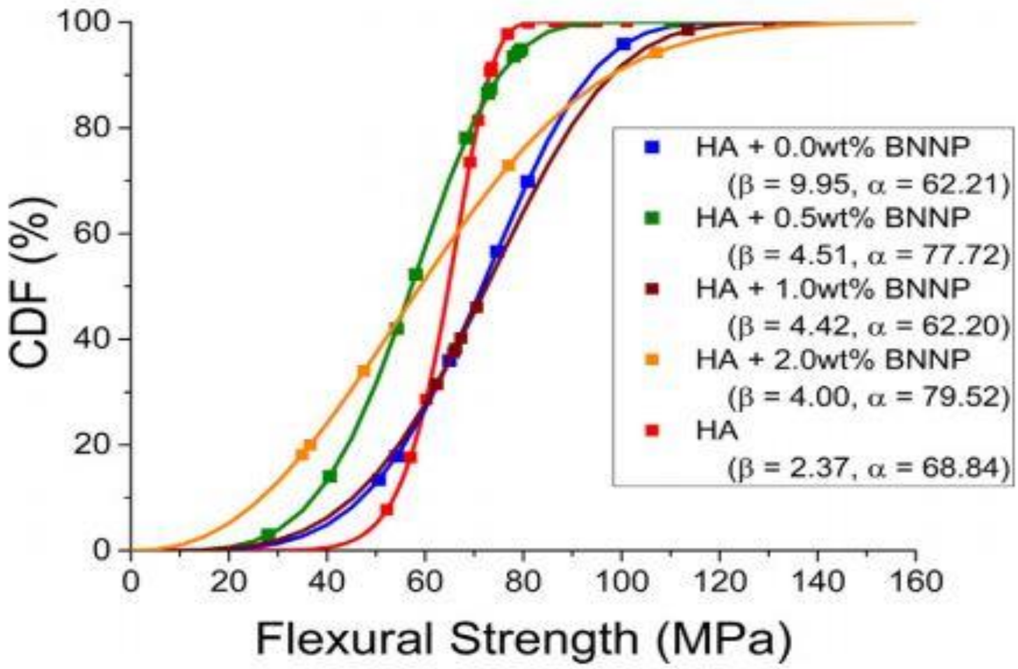

Figure 7.5b CDF for failure analysis [9]

3.5.2 EFFECT OF WEIBULL SCALE PARAMETER ON DISTRIBUTION

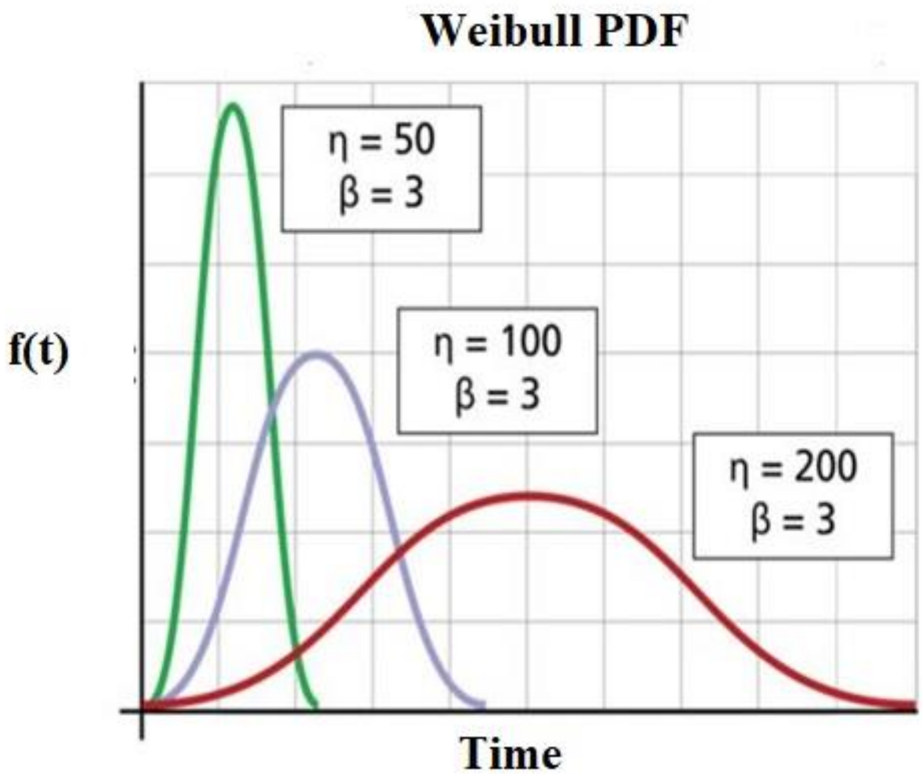

Figure 8.6 Effect of different β values on distribution [20]

Changing the value of scale parameter also has significant effect of curves. Increasing scale parameter by keeping shape parameter constant causes effect of stretching of probability density function.

- When value of ɳ increases while keeping other parameters same, the distribution gets stretched towards right with decrease in height.
- When value of ɳ decreases while keeping other parameters constant, the distribution gets stretched towards left with increase in height.
- Scale parameter ɳ has same units as data.

3.5.3 EFFECT OF LOCATION PARAMETER

The location parameter specifies the location distribution along x axis. Changing the value of ɣ causes the slide of distribution. If ɣ > 0, it slides to the right while if ɣ < 0, it slides to the left.

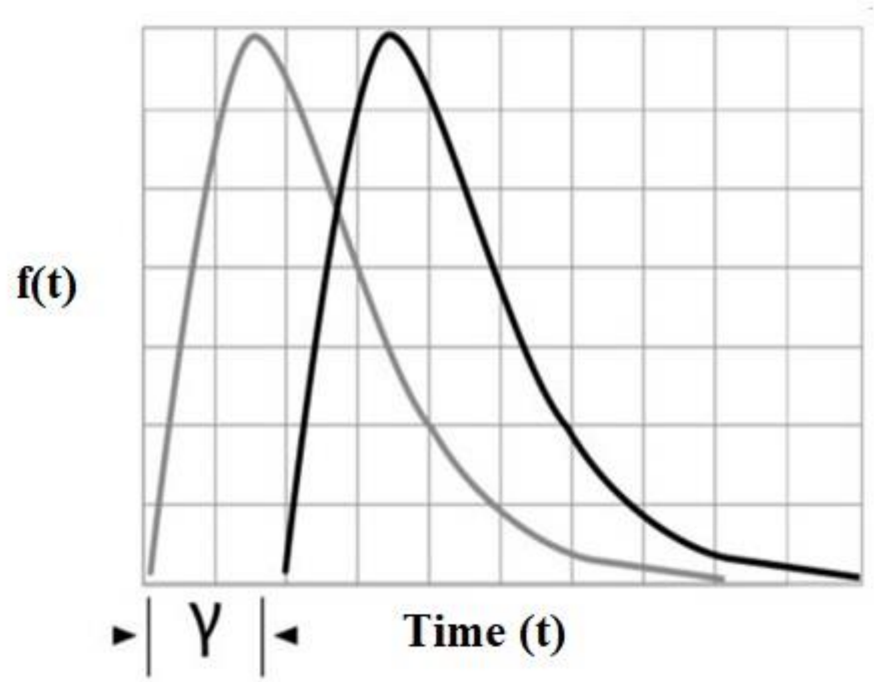

Figure 9.7 Effect of different β values on distribution [20]

- If the ɣ = 0, the distribution starts from origin.
- If ɣ > 0, the distribution starts to the right of origin.
- If ɣ < 0, the distribution starts to the left of origin.
- ɣ provides the information of earliest time of failure of specimen.
- ɣ has same units as a quantity expressed in data.

3.6 PARAMETER ESTIMATION

3.6.1 WEIBULL PLOT

One method of finding the parameters is plotting the probability. The method of plotting contains the first step in which times of failures are ranked in ascending order and obtain the median rank plotting. Reason of using median ranks is that, it has a specific confidence level of 50 % . Median ranks can be estimated from table as well as equation. The slope of the straight line passing through all points drawn according to ranks, gives shape parameter. In Matlab this plot can be drawn by a command x=wblplot(data).

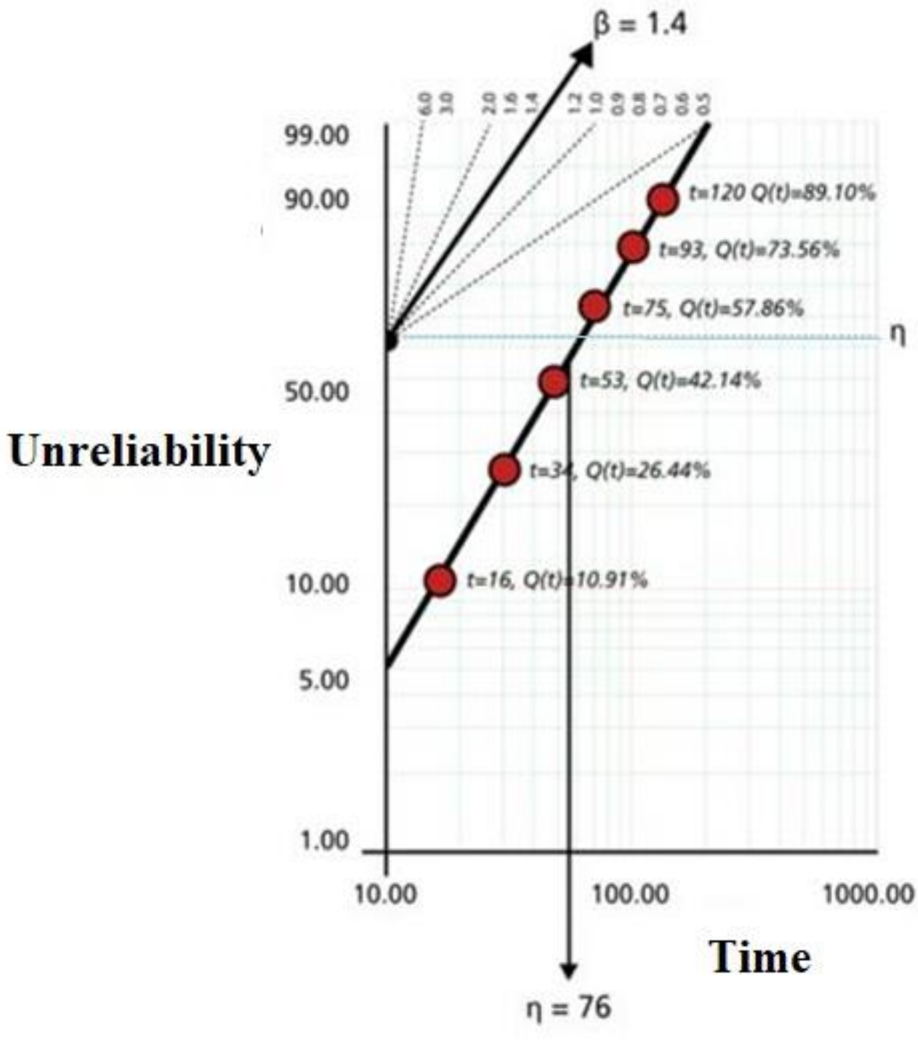

Figure 10.8 Weibull Plot

Drawing a horizontal line at 63.2 % probability gives scale parameter.

3.1.5 MEDIAN OF WEIBULL DATA

The median of Weibull probability distribution is given by;

$$\breve{T} = \gamma + \eta(\ln 2)^{1/\beta} \tag{7}$$

3.7 Effects OF PARAMETERS ON PDF

There are several methods that can be used for estimation of Weibull parameters. Some of these methods are; probability plotting, rank regression, maximum likelihood estimation and energy pattern factor method.

3.8 CALCULATED RESULTS AND PLOTS

Once Weibull parameters that fits the data are known, following results can be concluded.

- **Reliable time:** It is basically the probability that a specimen will operate without failure to a specific time.
- **Failure time Probability:** it is the probability that a specimen will bear a failure at particular time. This is also known as unreliability that is reciprocal of reliability of product.
- **Average life:** it is the average time for which a specimen can operate without failure.
- **Rate of Failure:** it is defined as the number of failures that can happen per specimen per unit time.
- **Reliability Period:** It is basically the warranty of the specimen defined as the suggested time for which the product's reliability will be equal to the set goal. For example, the estimated time of successful operation of product is 1 year for 90 % reliability.
- **Probability Plot:** It is plot consisting of failure probability over time. The probability plots uses linearized data distribution so, the form for one probability plot is different from the form of other probability plot.
- **Time Plot vs Reliability Plot:** it is plot consisting of reliability over time.
- **Probability Density Plot:** This is a plot of probability function known as probability density function.
- **Time vs Failure Rate Plot:** This plot consists of failure rate over time.
- **Contour Plot:** A graphical representation of the likelihood ratio equation solutions is known as contour plots. This type of plot is used to make a comparison between two data sets.

3.9 CONFIDENSE BOUNDS:-

Results obtained from Weibull analysis are based on observed data of a specimen and there is a level of uncertainty due to limited data points. Confidence bound can be said as two sided confidence or one sided confidence. Two sided bounds are used for indication that the value of interest is contained at specific confidence within bound. One sided confidence is used for the cases if the desired value of interest is above or below the lower and upper bounds respectively. Appropriate bound type is selected on the basis of application. For example a data analyst can suggest one sided an upper bound for the specimen failing probability and one sided lower bound for specimen reliability.

For example 90% lower and upper two sided bounds are 95% lower and upper one sided bounds respectively.

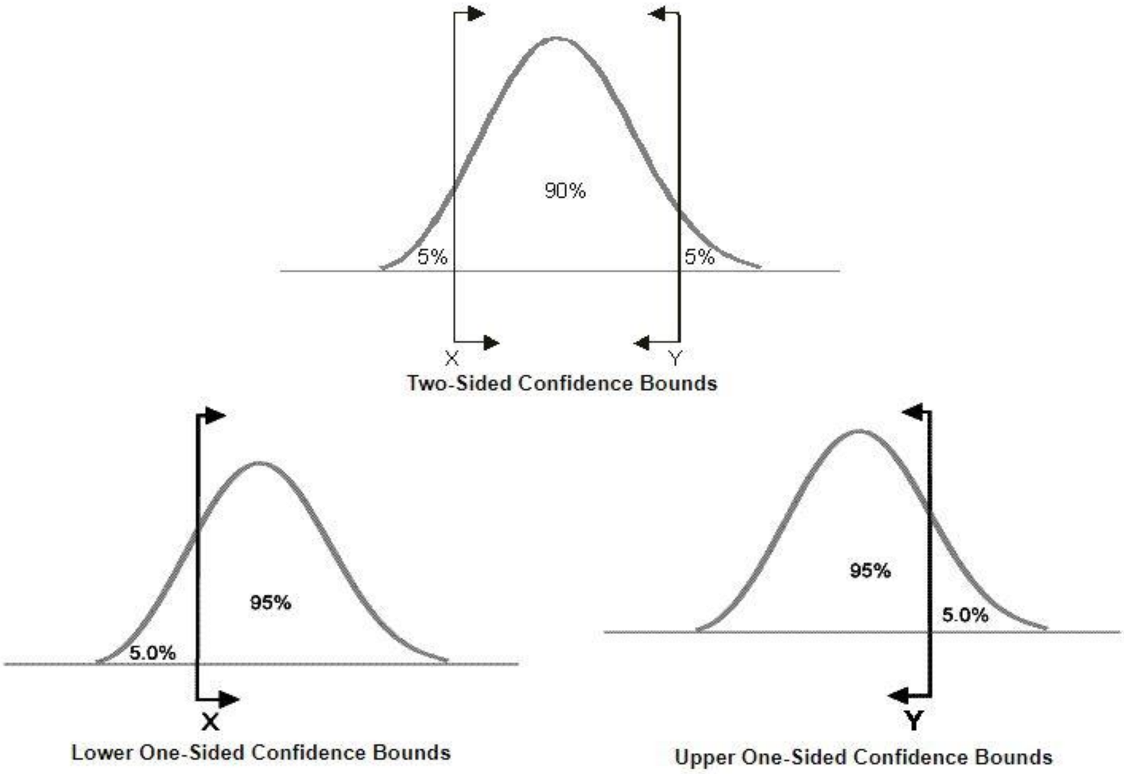

Figure 11.9 Confidence bounds [20]

CHAPTER 4: ARTIFICIAL NEURAL NETWORKS

This chapter consists of brief introduction of types of neural networks and their learning. It contains introduction of network architecture of both perceptron and multilayer perceptron. This chapter also demonstrates effect of input parameters and hyperparameters on results and training of NN respectively.

4.1 ARTIFICIAL NEURAL NETWORK (ANN):-

Artificial Neural Network (ANN) is an interconnected network of artificial neurons that can learn by supervised or unsupervised way. Each neuron that is also known as perceptron make simple input based decisions and provide information to other neurons that are present in next interconnected layer. Artificial neural network can imitate any function even if there are finite number of discontinuities and practically answer questions. Neural networks can be categories as shallow or deep on the basis of architecture. Increasing number hidden layers make the network capable of describing nonlinear behaviors.

Neural network consists of neurons present in different layers. The layers cann be;

- Input Layer:- This layer accepts the independent values of input parameters. The number of input neurons depend on the number of input parameters.
- Hidden Layer:- These layers are present in between input and output layers. Hidden layers receive information, process it and transmits to output layer.
- Output Layer:- this layer generates the prediction.

Shallow Neural Networks have only one hidden layer while Deep Neural Network has two or more than two hidden layers. Courville, Goodfellow and Bengio described that even shallow neural networks can also tackle complex problems, but deep learning can improve the accuracy of results as more neurons and hidden layer enhances the capability of catching nonlinear trends. Most of the well performing neural networks have 3 – 10 layers and if the quantity of layers is increased further, it decreases the prediction capability. A comparison between shallow and deep learning neural network is discussed by J. H. Lee [10] and the architectures are shown below in fig 4.1.

.

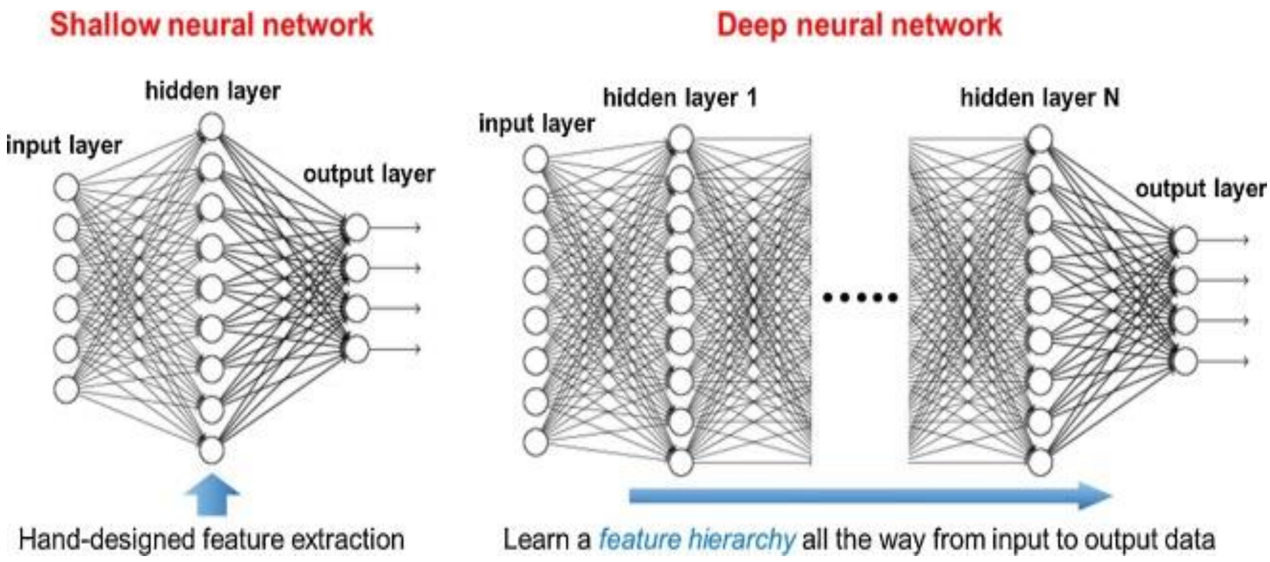

Figure 12.1 Architecture of shallow Vs deep learning ANN

4.2 ARTIFICIAL NEURAL NETWORK CONCEPTS

4.2.1 INPUTS

The data from the source is fed to the neural network through its input layer. The purpose of training neural network can be anything, like in case of this study, the purpose is wind speed estimations and predictions. Inputs to the neural network are real values and each value is fed to a neuron in the input layer.

4.2.2 TRAINING

Known output values from the past data is used as targets for each input example. These targets trains the neural network and helps in finding the performance of neural network.

4.2.3 OUTPUTS

Neural networks have two types of output. One type of output is on each neuron of hidden layers and second is the final output at output neurons. Each output is basically a real value or Boolean decision by the output neurons in last layer.

4.2.4 NEURON

Neuron is the basic component of neural network that is also known as perceptron. It accepts the input and generates output. It uses simple summing function.

Each neuron accepts real values of input and passes the neurons output through an activation function for final output. The activation functions are sigmoid transfer functions, TanH and modern ReLu activation functions. Activation function helps in generating the output in acceptable range and introduces the nonlinear behavior for training.

4.2.5 WEIGHT SPACE

Each neuron accepts the input, have a numeric weight. The weight defines the neurons output that become the final output after passing through activation function. So, both weights and activation function are responsible for defining the output. Before starting training the weights are provided with initial values by an initialization function, known as weight initialization and then neural network is trained by fine tuning of initial weights to discover the optimal set of new weights having better performance to generate most accurate predictions.

4.2.6 FORWARD PASS IN NEURAL NETWORK

In forward pass step, the inputs are taken and passed through network and allows each neuron to react on the inputs. Output of each neuron is passed to next connected neurons until the network generates the output.

4.2.7 ERROR FUNCTION

Error function defines the difference between the outputs of network and actual values. The aim of the training is to minimize the difference between current model and actual output.

4.2.8 BACK PROPAGATION

In order to get the optimal weight values for better performance, back pass plays an important role. The propagation performs a backward pass moving from prediction to the neuron that generated the prediction. Tracking derivatives of activation functions in each successive neuron helps in making performance better in each iteration. This mathematical process is known as Gradient Descent method.

4.2.9 BIASNESS AND VARIENCE

During the process of training it is important to make a well balance between biasness and variance.

Bias tells that how well the designed model fits input data with minimum error and better performance while variance explains the generalization limit of neural network. Variance is basically the behavior of neural network for unknown inputs, these inputs were not available for training. Bias value moves the activation function left, right, up and down.

4.2.10 HYPER PARAMETERS

Hyperparameter settings effect the overall structure and performance of neural network. Tuning these hyperparameters is a basic way of building a network that produces accurate predictions. The hyperparameters are activation function, number of hidden layers and epochs.

4.3 BASIC ARCHITECTURE OF NEURAL NETWORK

4.3.1 PERCEPTRON

Perceptron is an algorithm for binary classification that is designed after human brain. It has simple structure but can learn and solve complex problems.

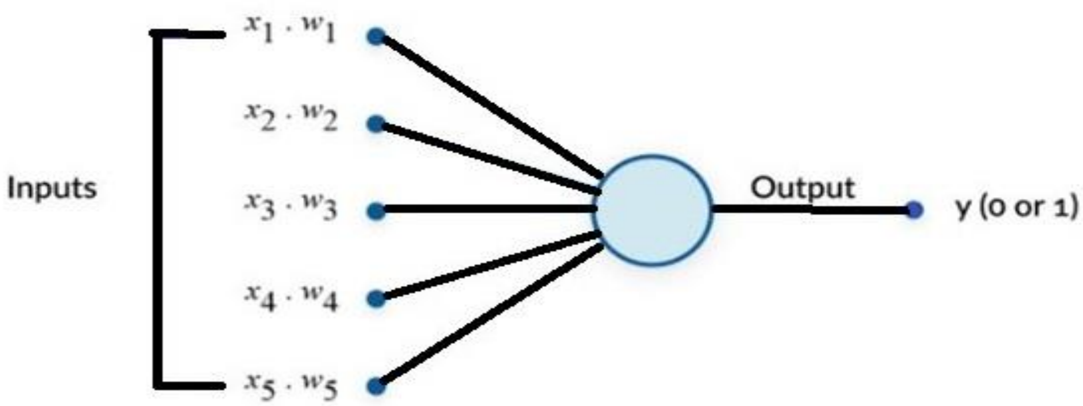

Figure 13.2 Architecture of Perceptron

Where **x** are the inputs, **w** are weight values and **y** is output.

4.3.2 MULTILAYER PERCEPTRON

Multilayer Perceptron consists of multiple layers having perceptrons. This type of neural network can answer complex problems more effectively. The input signal reaches the perceptrons in first layer and then passes on to the next layers. At least one hidden layer is part of this neural network type. This type of Neural Network has many applications like Zlatan Car [15] used it for modelling of spread of Covid-19 virus.

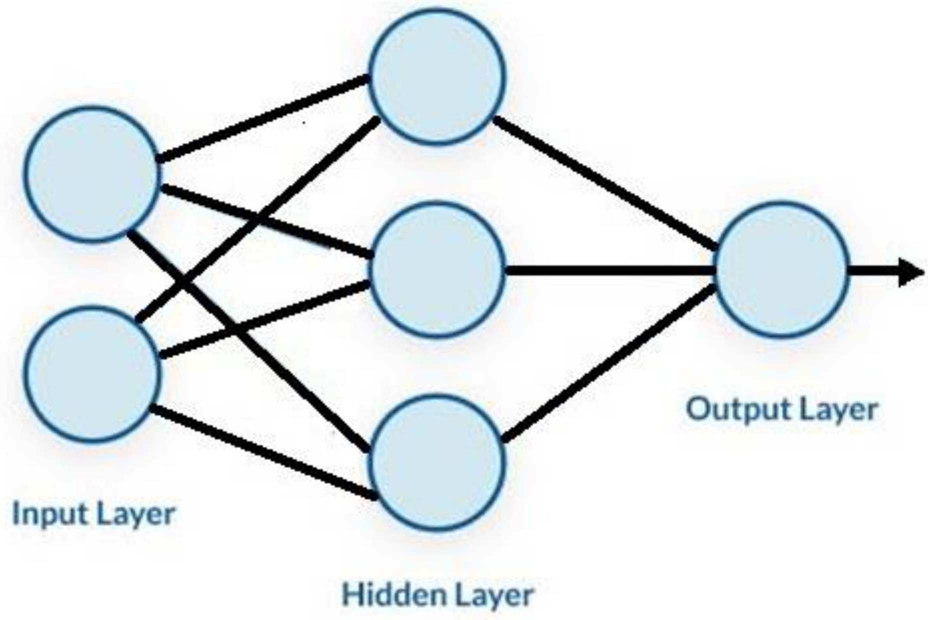

Figure 14.3 Multilayer Perceptron

4.4 LEARNING OF PERCEPTRON

Perceptron learns by following steps;

- Inputs are given to the perceptron in input layer then after multiplying inputs with weights, sum is computed.
- Adds bias which causes the shifting of output function to up, down, left or right.
- Passes the results from activation function. Commonly used activation function for perceptron is step function.
- The results of step function are final outputs which are compared with target values.

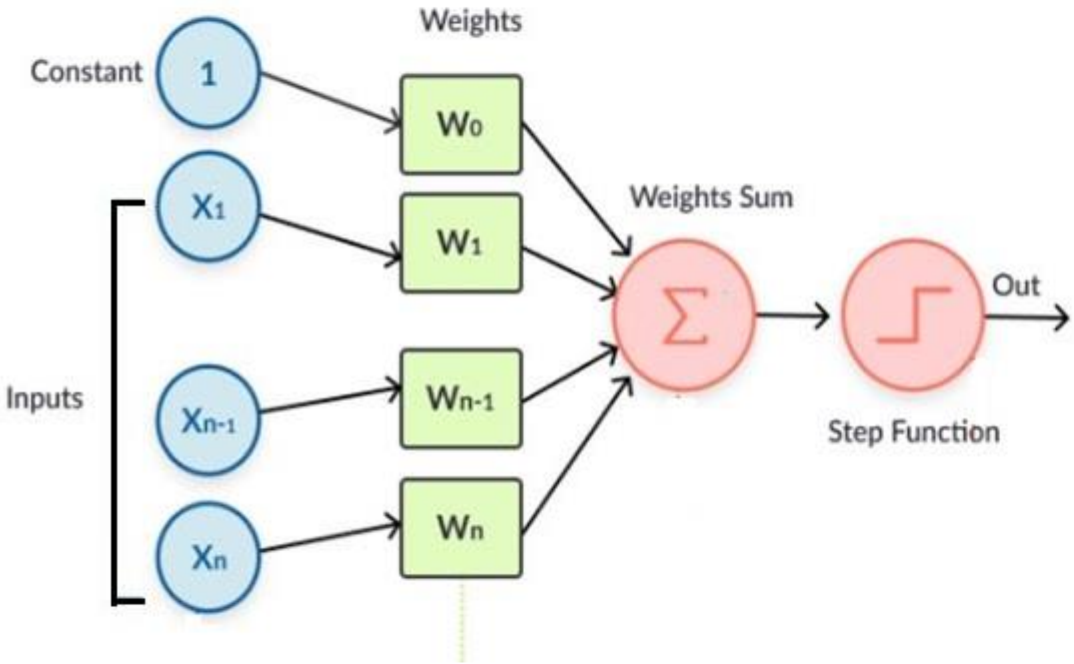

Figure 15.4 Learning of perceptron

4.5 DEEP LEARNING NEURAL NETWORK

Multilayer perceptron network becomes deep learning system by adding more hidden layers. Multilayer perceptron is close to modern neural network systems.

- **Activation Functions:-** there are number of activation functions that produces real values rather than Boolean values as outputs unlike the traditional perceptrons. Deep learning neural networks are more flexible is terms of other details of hyperparameters. These parameters are training iterations, weight in initialization functions, regularization and normalization etc.
- **Backpropagation:-** A backpropagating neural network makes an iterative back pass to estimate optimal weight values that generate accurate prediction.
- **Advanced Architectures:-** Advanced full neural networks have a variety of architectures that can help in solving specific problems. Some modern neural network architectures are Recurrent Neural Network, Convolutional Neural Network and Generative Adversarial Neural Networks etc.

4.5.1 BACK PROPAGATION

Once the neural network is designed with hyperparameters and initial weights, the network performs forward pass for initial predictions. The error determines how far is the designed model from real values. There are many algorithms that minimizes the error function. Most effective algorithm converges fast and finds weights that generate smallest error. High computational ability is also a quality of feasible training algorithm. Backpropagation is an algorithm that finds weights very quickly even if there are millions of weights.

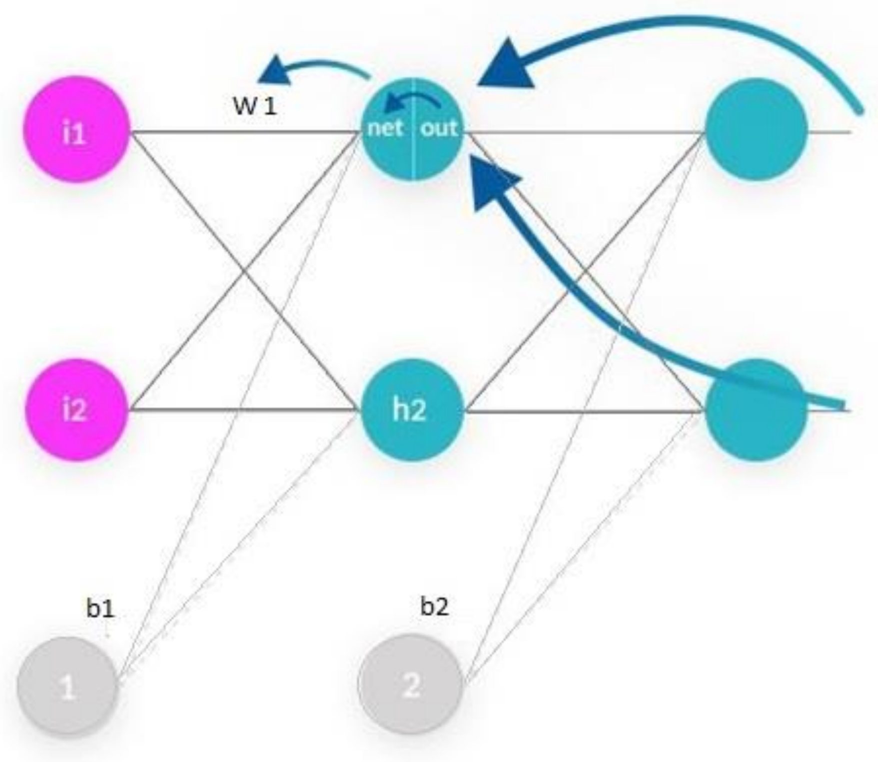

Figure 16.5 Backpropagation in learning

- **Forward Pass :-** As discussed, in forward pass information moves from input neurons to hidden neurons and then to output neurons to make prediction once after the weights are initialized. Output is defined by summing operation at neuron and activation function.
- **Error Function :-** Error function is computed by calculating the difference between predicted value and target value.
- **Gradient Descent Method :-** Gradient is most popular backpropagation algorithm which calculates how much output value is effected by each weight. In order to find this gradient descent method calculates the partial differential of error function with respect to its weight while going back from output neuron to hidden neurons. This provides the complete information of contribution oof each weight in error. The result obtained from back propagation is new weights that minimizes the error.
- **Weight Updates:-** In weight updating process, previous weights are replaced by new weights after every sample in the training set described by Marta Kolasa [11]. The change in weight values is according to calculated error gradient, the initial weight value is due to weight initialization function, that can be random function as discussed by Giulia

Napolitano [13]. This is incremental learning. While in case of batch training, a batch of samples is passed in forward direction. Iterations is also an important hyperparameter defined as number of batches passed for training. Running entire training set of specific size through backpropagation is called an epoch.

4.5.2 NEURAL NETWORK ACTIVATION FUNCTIONS

Activation functions are main part of deep learning architectures. Activation function controls the computational efficiency, converging ability and output after multiple iterations.

An activation function consists of a mathematical model that defines the final output of perceptron or neuron. Activation function takes input from neuron and produces output between zero and one or -1 and 1. Classic activation functions used had step function, sigmoid function and tanh function. New activation functions improved efficiency with ReLu and sigmoid tanh.

4.5.2.1 IMPORTANCE OF ACTIVATION FUNCTION

The values that are fed at input of are real values. Weight at each neuron is multiplied by input and passed through activation function. The output from activation function of a neuron is input for all other connected neurons in next layer. For complex problems, artificial neural networks completely rely on activation functions that are nonlinear. The derivative of activation function helps the network to learn in back propagation.

4.5.2.2 COMMONLY USED ACTIVATION FUNCTIONS

Sigmoid Activation Function :- this activation function has smooth gradient and has output value in between zero and one. It has a problem known as vanishing gradient problem for very high and low parameter values. As a result of this problem network reaches the prediction very slowly.

- **TanH Activation Function :-** this activation function is zero centered, it is commonly used for problems in which input values are strongly negative, positive and neutral.
- **ReLu Activation Function :-** this activation function has high computational efficiency but it is unable to process zero and negative data values.
- **Leaky ReLu Activation Function :-** this activation function covers the weakness of ReLu. It has small positive slope in its small negative area part. It can process zero and negative values.
- **Parametric ReLu Activation Function :-** this function allows the the learning of negative slope. Inn back propagation it is effective to learn for zero and negative inputs.
- **Softmax Activation Function :-** This activation function is a special activation function for output neurons. It normalizes the output between 0 and 1 and provide a probability estimation that input belongs to specific class.

4.5.3 BIAS VALUE

In artificial neural networks, bias has two explanations:

- Bias neurons that gives bias value and is part of structure.
- A statistical concept that how good neural network is performing during predictions based on training samples.

Bias neuron is present in each layer with a value. This value shifts the transfer function left, right, up or down. Without bias, each neuron takes the input value and after multiplying with weight it is passed through activation function. In such case it is not possible to take a zero value and produce output two which can be necessary in some cases.

4.5.3.1 BIAS VS VARIANCE

In order to understand the bias and variance we have to consider the set of data made by dividing total data. First set is training data set, second one is validation data and last one is testing data.

- **Training Data Set :-** This data set is group of examples fed to the neural network for training.
- **Validation Data Set :-** This data set consists of examples that are not seen. It is used to test network for its performance. Network keeps on learning during validation.
- **Test Data Set :-** This data set consists of examples that are unseen and used to check for network performance. Neural network do not learn during testing.

4.5.3.1.1 BIAS

Bias tells about the performance that how well the neural network is performing. A high bias means the artificial neural network is not able to generate accurate outputs even tested on the examples it trained on.

4.5.3.1.2 VARIANCE

Variance describes the behavior of neural network for unseen examples. High value of variance defines the large prediction error by artificial neural network for unseen examples.

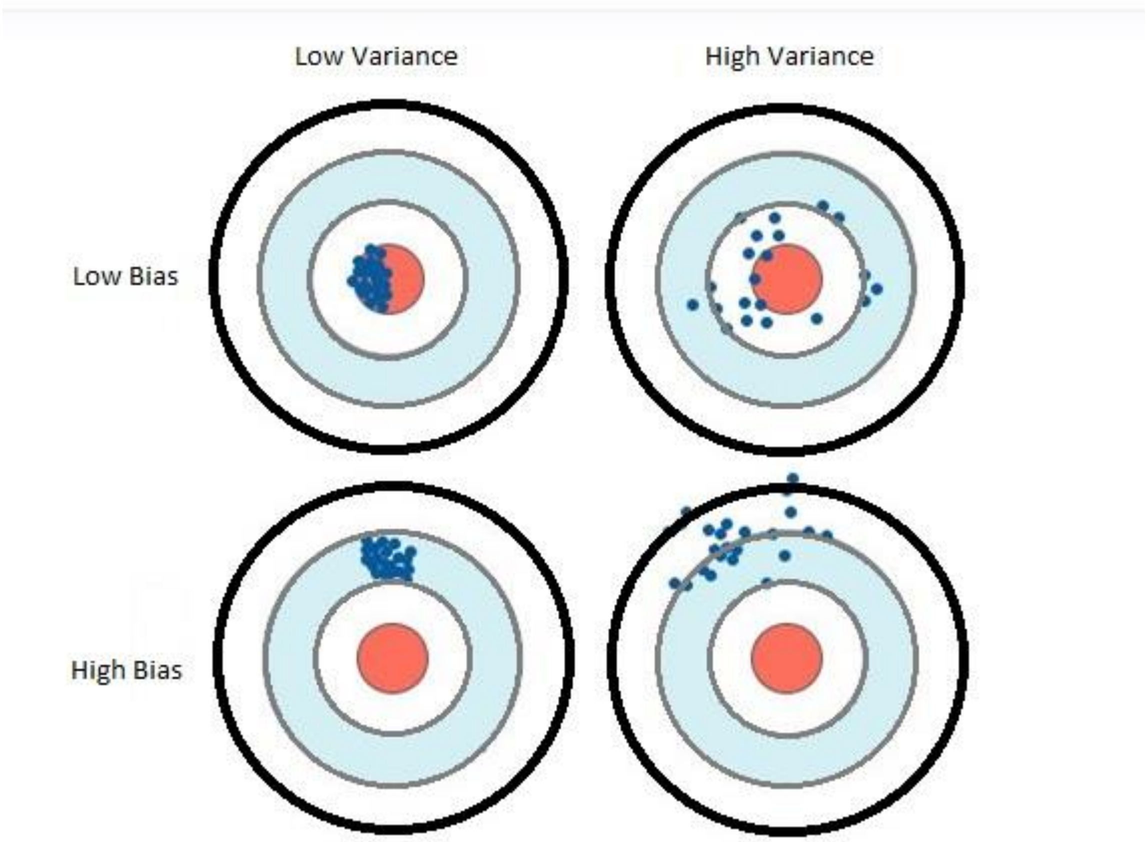

Figure 17.6 Bias and Variance

4.6 OVER FITTING AND UNDER FITTING OF NEURAL NETWORKS

Over fitting is a problem that occurs when artificial neural network is good at learning on training data set but weak in generalization to additional data that is unseen. This is characterized by high variance and low bias.

Underfitting is a problem in which an artificial neural network is not able to predict for data on which it is trained. Its results for validation also has huge error. This error is defined as high bias and high variance.

4.6.1 METHODS TO AAVOID OVER FITTING OF NEURAL NETWORK

The common methods to avoid the problem of over fitting are as under:

- **Retraining the artificial neural network :-** Retraining means running the same algorithm on same neural architecture for same data but with different weight values

- **Training Multiple Neural Networks :-** Training multiple neural networks mean, training several network models in parallel having same structure but different weights. The output in this case will be the average of two networks.
- **Stopping Neural Network Training Early :-** In this type of solution, practitioner observes the validation error after each iteration and stops when network start to become over fit.
- **Applying Regularization :-** Regularization means adding a term to error function. This action results in decreasing weights and biases, smoothing outputs anndd prevents the network from over fitting.

4.6.2 METHODS TO AVOID UNDERFITTING OF NEURAL NETWORK

Underfitting of neural network can be avoided by following means:

- **Increasing Inputs or Number of Neurons :-** Underfitting can be avoided by adding more hidden layers or more number of neurons in each layer. These neurons produce more complex predictions and improve the results.
- **Increasing Training Data or its Quality Improvement :-** Under fitting problem can be avoided if more data for training is fed to the artificial neural network and these data points present better variance in real population.
- **Dropout :-** In this approach some percentage of neurons are killed at every training iteration resulting in removing some learned information. This reduces the risk of overfitting.
- **Decreasing Regularization Parameters :-** Over doing regularization can cause such problem. Selection of optimal regularization parameter is suggested to avoid underfitting.

CHAPTER 5: METHODOLOGY AND RESULTS

This chapter demonstrates the adopted methodology for prediction of wind speed and its results.

5.1 METHODOLOGY

This study suggests an approach of Artificial Neural Network (FFBP NN) speeds on hourly, daily, weekly, monthly and yearly basis. Input parameters that are chosen are temperature, pressure and humidity while target values are of wind speed. Trainlm is used as training algoritm and learngdm is used as adoptive learning function. Different hidden layers are used for different problem. Weight initialization function that is used is initwb. The performance function used with trrainlm are mean square error (mse) and sum squared error (sse).

Second type of used neural network is Nonlinear Autoregressive Neural Network (NAR NN). This neural network is used to predict future wind speed values. NAR NN requires a delay to be defined early in start. The default training algorithm is "trainlm".

This study is based on statistical methods for wind speed estimation, which develops a relation between input and output values. Statistical methods perform by a technique of input – output mapping. Conventional alternative of statistical method is physical methods, which use predefined mathematical model.

5.1.1 DATA TYPE

Data obtained for wind speed is not steady. After applying some checks, it was clear that data is neither normally distributed nor from Weibull distribution. Matlab is used for checking data type. For Weibull distribution a command "Wblplot(x)" is used, which plot that data according on Weibull plotting graph. If all data points comes along straight line then data belongs to Weibull distribution, as shown in fig 5.1.

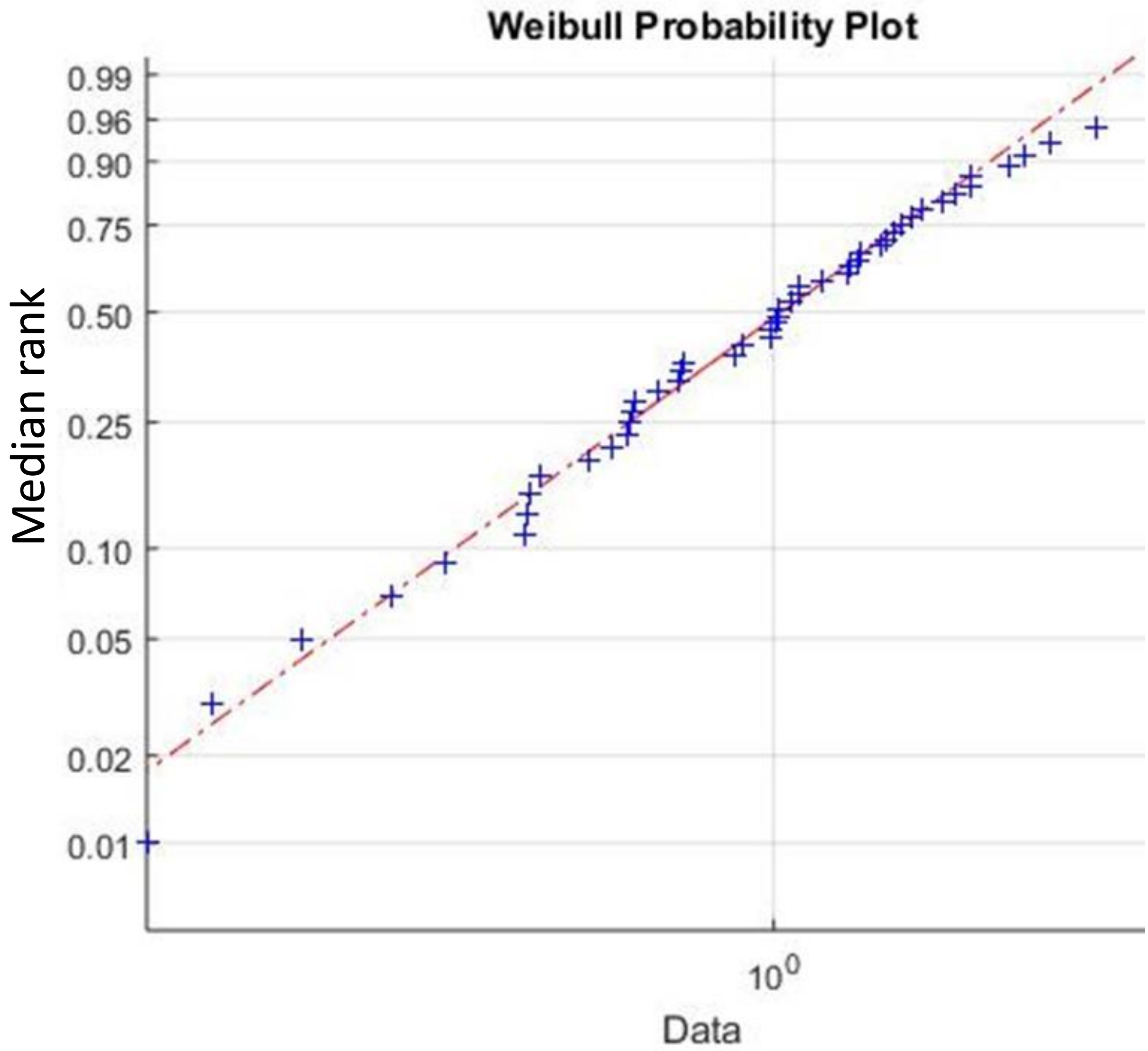

Figure 18.1 Weibull plot of randomly generated Weibull data

As above figure shows that all data points lie on same straight line which confirms that data belongs to Weibull distribution. This data is generated by **r = wblrnd(1.2,1.5,50,1).** This command generates a vector r containing 50 random numbers from a Weibull distribution with parameters A = 1.2 and B = 1.5.

But the results of wind speed data were not like this. Figure 5.2 shows the wblplot of wind speed data collected from site.

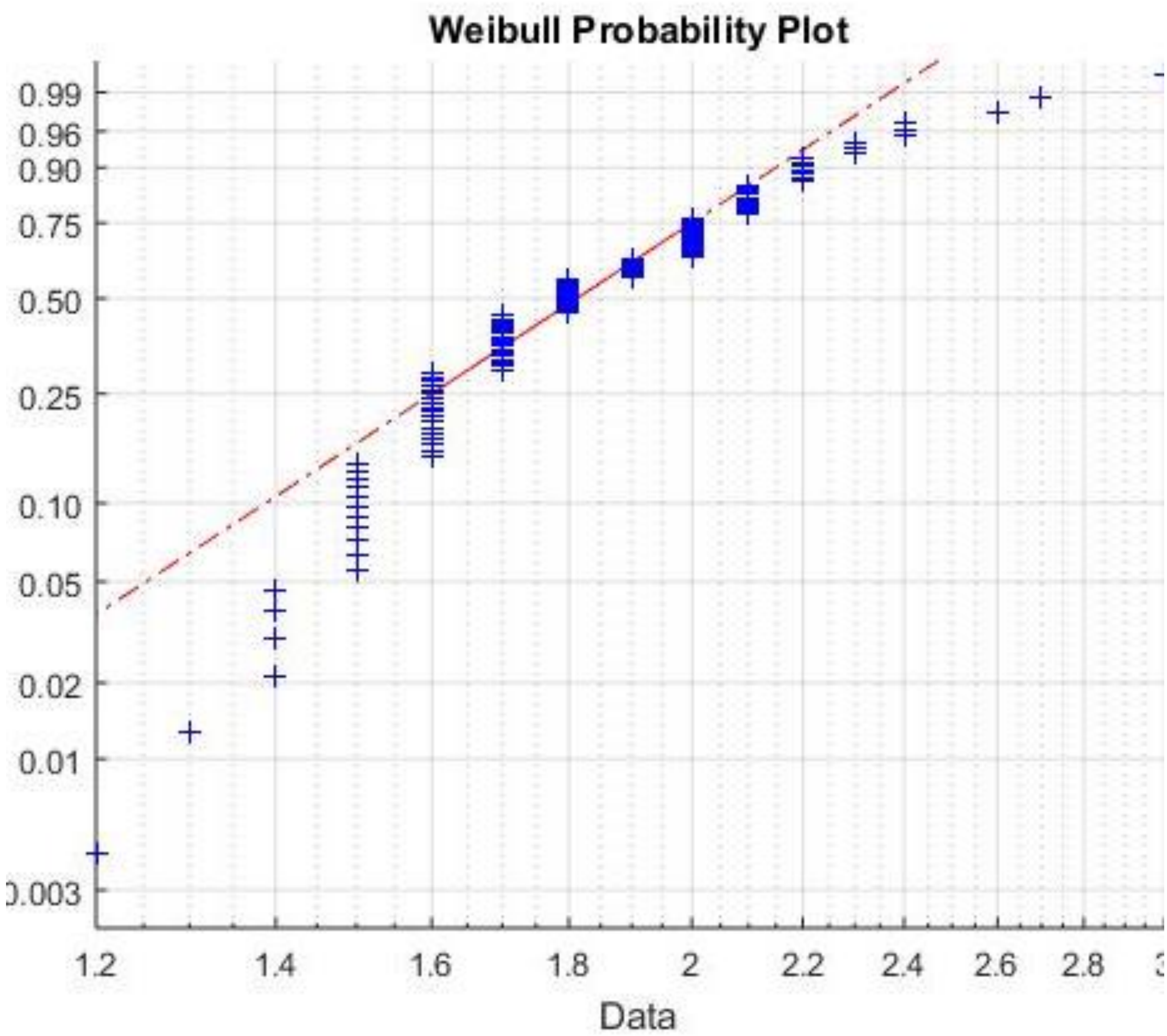

Figure 19.2 Weibull plot of wind speed data

5.1.2 NEURAL NETWORK DESIGN

5.1.2.1 FEED FORWARD BACK PROPAGATION NEURAL NETWORK

Design of Neural network involves selection of Neural Network type and Hyperparameters. FFBP NN is used for wind speed estimations based on three input parameters. The neural network is trained by supervised learning. FFBP NN consisted for three input neurons and one output neuron because of inputs and targets. The view of neural network is shown in figure 5.3

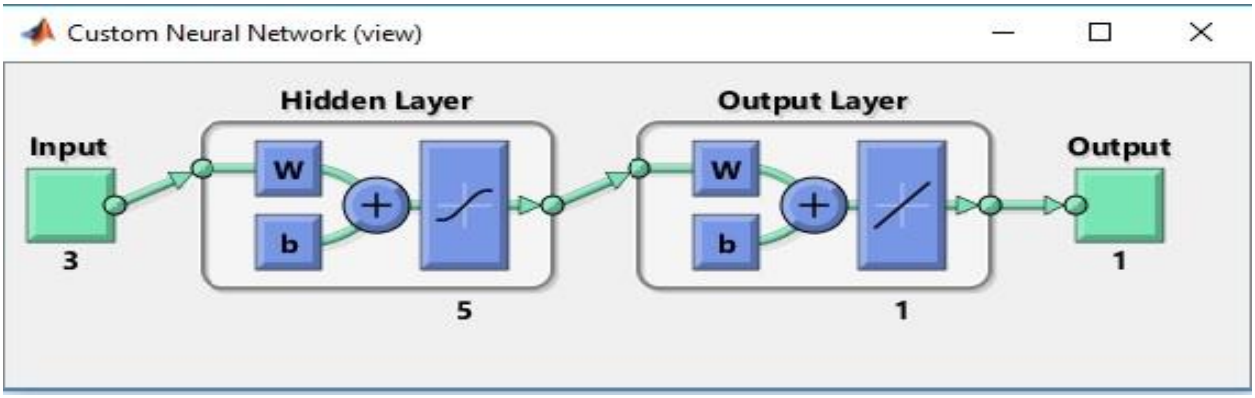

Figure 20.3 Architecture of Neural Network

Figure 5.3 demonstrates three input neurons and one output neuron. There are five hidden layers that are fully connected. Linear activation function is used for output layer while nonlinear activation function is used for hidden layers to catch the complexity of function.

5.1.2.2 HYPER PARAMETER SELECTION

Selection of right hyper parameters are very important for training. The designed neural network consisted of nonlinear logsig activation function for hidden layers and purelin, that is a linear activation function for output layer. The other hyper parameters which are responsible for stopping training are given in below table 5.1;

Table 5.1: Hyper parameters

DESCRIPTION OF PARAMETER	VALUE OF PARAMETER
Maximum number of epoches	6000
Momentum (mu)	0.01
Momentum decrease (mu_dec)	0.01
Momentum increase (mu_inc)	10
Maximum momentum (Mu_max)	10^{100}
Goal	0
Minimum gradient (Min_grad)	10^{-5}
Training time	inf
Maximum fails (Max_fail)	6

5.2 RESULTS

The trained neural network was tested for different unseen data types which are discussed below. Initially network was tested for yearly inputs of temperature, pressure and humidity. The network was trained on monthly data but tested for yearly data. The results are shown in figure 5.4 and mean absolute percentage error was 7.7 %.

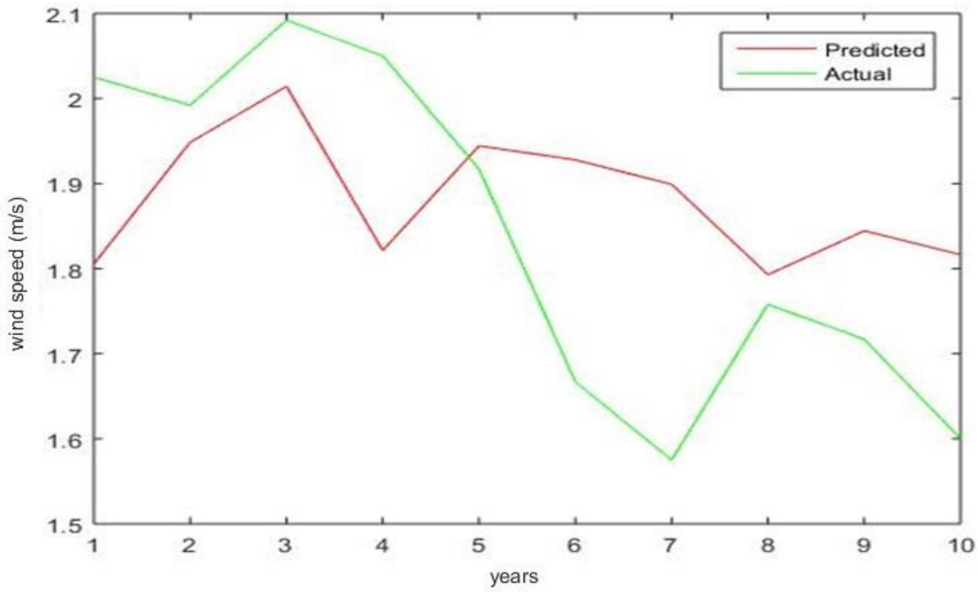

Figure 21.4 Yearly wind speed predictions

Tale 5.2: Real Vs Predicted values of FFBP NN

Actual Value	**Predicted value**	**% Error**
2.01	1.8	10.4
2	1.95	2.5
2.1	2	4.7
2.05	1.8	12.19
1.9	1.95	2.5
1.62	1.925	15.8
1.58	1.9	16.8
1.75	1.8	2.7
1.7	1.85	8.3
1.6	1.81	11.6

Same FFBP NN when trained for daily data values the results are as under;

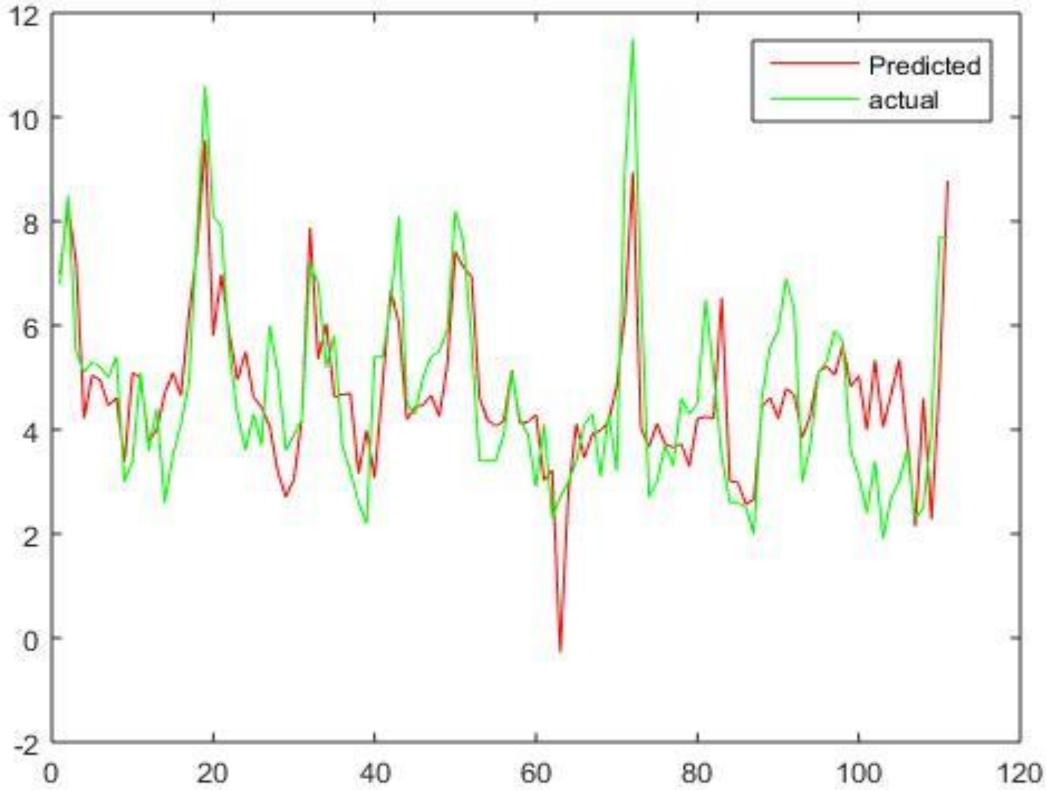

Figure 22.5 Daily wind speed estimation by FFBP NN

Number of input neurons are dependent on the number of input parameters considered in supervised learning. Wind speed data is not a steady data and there are variety of parameters that effect its values. Wind speed is caused by the motion of air molecules between two regions having difference of pressure. Pressure gradient is the driving force for air molecules, and they move from a region having high pressure value to low pressure value. Temperature effect the pressure because hot air moves upward creating a wake, that is filled by air in surroundings. Hence the ANN with two input parameters consists of Temperature and Pressure values only while ANN with three parameters consists of Temperature, Pressure and Humidity. Table 5.5 shows error comparison between both ANNs and it is clear that considering an additional parameter improved the results.

Tale 5.5: Error comparison between NNs trained on two and three parameters

Predicted data type	**Mean Absolute Percentage Error**	
	Two Parameters	**Three Parameters**

Yearly Prediction	9.04%	7.7%
Daily Prediction	10.3%	8%

5.1.2.2 NONLINEAR AUTOREGRESSIVE NEURAL NETWORK

NAR NN can be trained to predict time series from past data. Its arguments contain feedback delay, hidden layer size and training function. Once the network is trained it can be used to predict future values. The architecture of NAR NN is shown in figure 5.6.

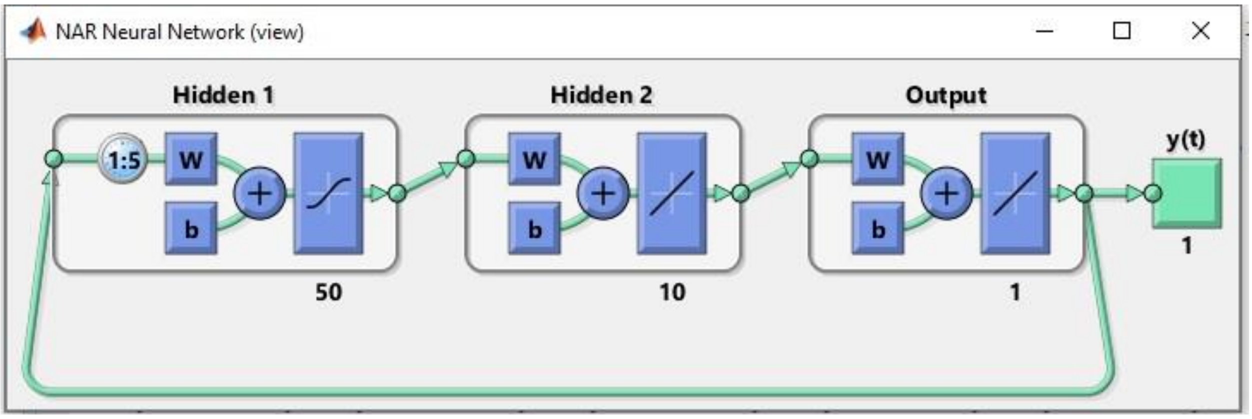

Figure 23.6 NAR NN architecture

Fig 5.7 shows the wind speed prediction using NARNet trained on monthly wind speed real data with mean absolute percentage error of 19.8%. Fig 5.8 shows the weekly wind speed prediction using NAR Net with 16.3% mean absolute percentage error.

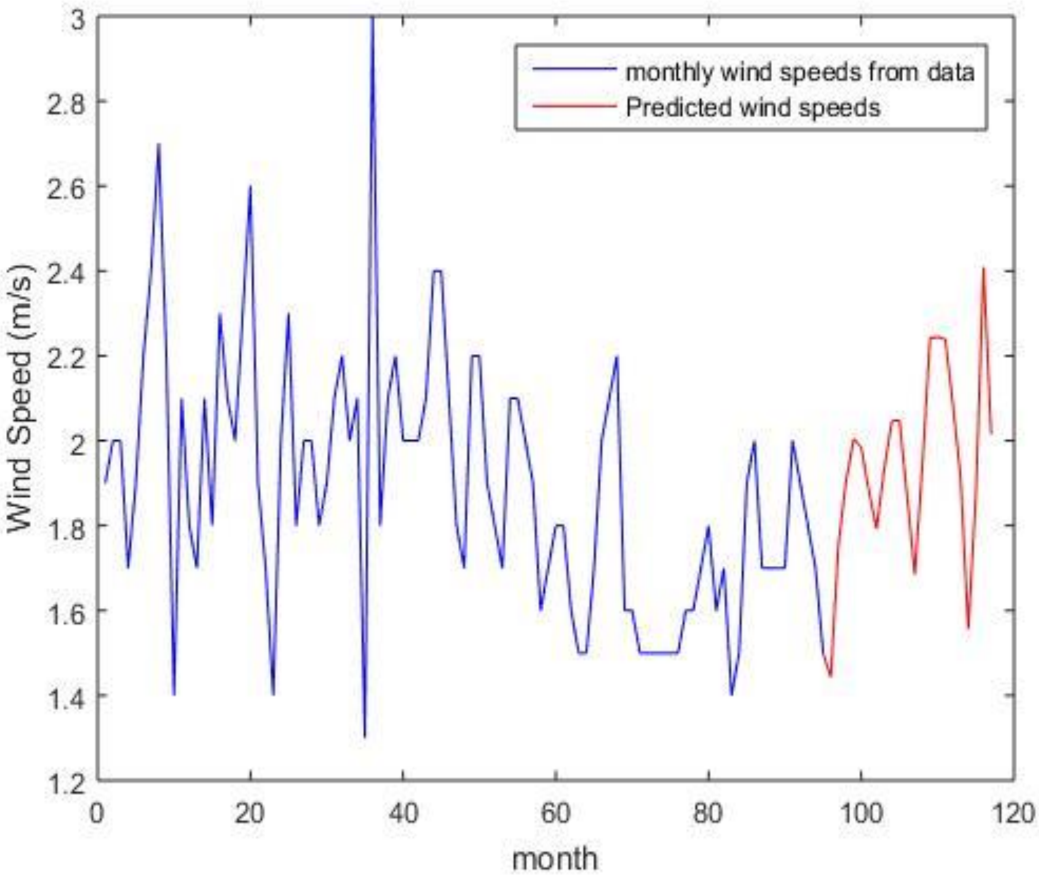

Figure 24.7 Monthly wind speed predictions by NARNet

When NAR Net is trained for prediction of weekly data the error was 16.34%.

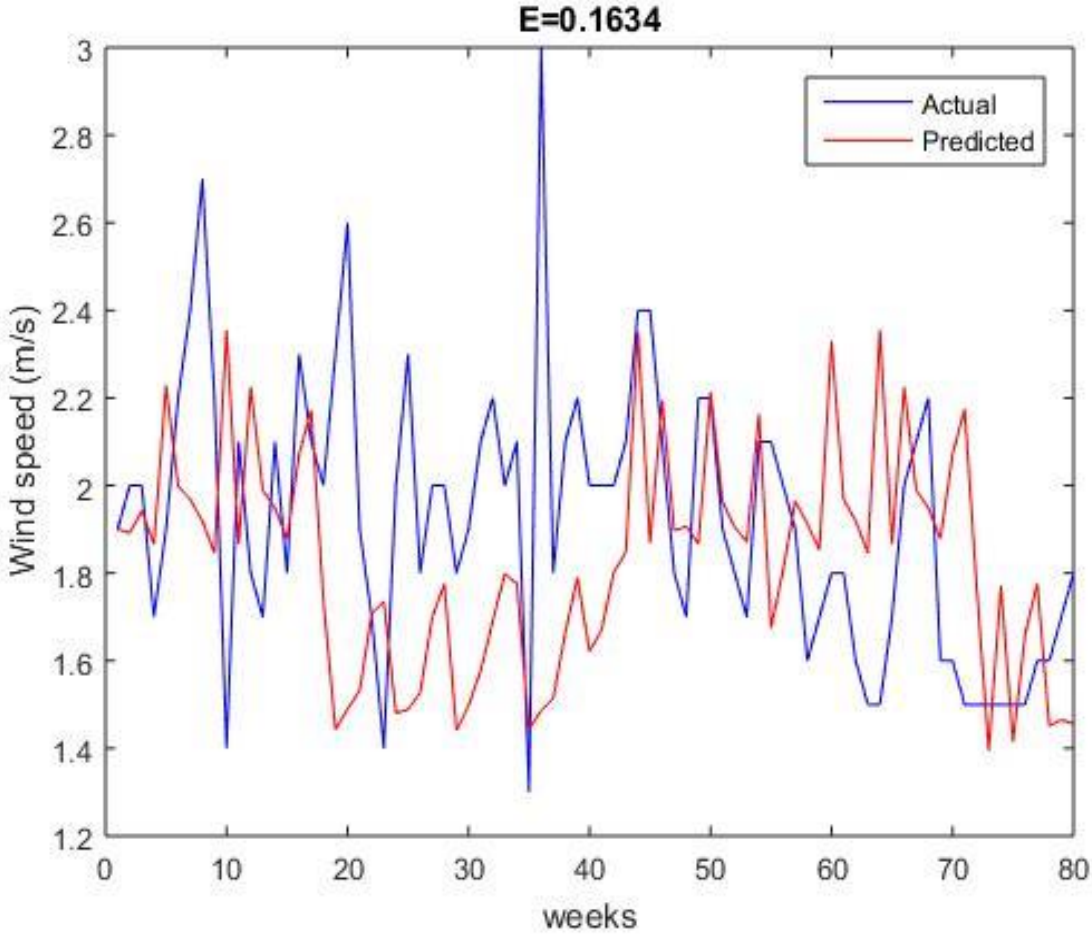

Figure 25.8 Weekly wind speed predictions by NAR Net

CHAPTER 6: CONCLUSION AND FUTURE WORK

This chapter consists of conclusion of the study and future work depending on detailed literature review. Also some other areas of application of ANN are part of this chapter.

6.1: CONCLUSION

This study briefly describes the use of FFBP NN for wind speed estimations and NAR NN for wind speed predictions of future intervals.with reasonable accuracy. Prediction of wind speed plays an important role in weather forecasting, aviation and maritime operations. Wind speed data is also required for site selection of wind turbines. There are many methods that are conventionally used for wind speed predictions and majority of them are physical methods. The most common example of such method is Weibull analysis. These physical methods require predefined mathematical models. Real time data of wind speed is not a steady data, also its very possible that it will not follow any distribution. Such problems need an alternative statistical approach that work on the basis of input output mapping. ANN is best statistical approach for such data types.

Comparative study of physical and statistical approach showed how ANN overcomes the limitations of physical methods and give reasonable output independent of data distribution and finite discontinuities. The accuracy of the predicted data depends on the input parameters. Wind speed is effected by large number of parameters, most common of them are temperature, pressure gradient and weather conditions of the region. This study is an improvement in existing study which contained only temperature and pressure as input values and neglected the humidity. While considering humidity showed improvement because dense air has greater inertial effect and humidity effects the density of air.

6.2: FUTURE WORK

As discussed, the accuracy of results depends on the precise selection of input parameters. When we are concerned for long term wind speed predictions at greater height, Rossby waves, that are the inertial waves have greater effect. It can be considered for further improvement in results.

ANN has lot more industrial applications where we cannot use physical methods eg in mill settings and predictions of different parameters depending on input values. These are best future study area for ANN implementation

REFERENCES

[1] Kuo, P.-H.; Huang, C.-J. “An Electricity Price Forecasting Model by Hybrid Structured Deep Neural Networks”, Sustainability 2018, MDPI, Open Access Journal, vol. 10(4), pages 1-17, April 10, 1280.

[2] De Filippo, Lombardi, Milano, “User-Aware Electricity Price Optimization for the Competitive Market”. Energies 2017, 10, 1378

[3] Cincotti, S.; Gallo, G.; Ponta, L.Raberto, M, “Modeling and forecasting of electricity spot-price”, Computational intelligence vs classical econometrics. AI Commun. 2014, 27, 301–314.

[4] Jiang Y, Song Z, Kusiak A. “Very short-term wind speed forecasting with Bayesian structural break model”, Renew. Energy 2013, 50, 637–647.

[5] Bouzgou, H. “A fast and accurate model for forecasting wind speed and solar radiation time series based on extreme learning machines and principal components analysis”, J. Renew. Sustain. Energy 2014.

[6] Fortuna, L.; Nunnari, G.; Nunnari, S. “A new fine-grained classification strategy for solar daily radiation patterns”, Pattern Recognit. Lett. 2016, 81, 110–117

[7] Yu-Cai Zhang, Hui-Qin Zhao, Wenchun Jiang, Shan-Tung Tu, Xian-Cheng Zhang, Run-Zi Wang, “Time dependent failure probability estimation of the solid oxide fuel cell by a creep-damage related Weibull distribution model”, International Journal of Hydrogen Energy 2018

[8] Chin-Diew Lai, D. N. Pra Murthy, Min Xie “Weibull Distributions and Their Applications”, 2006 Springer Handbook Of Engineering Statistic (pp.63-78)

[9] Trevor G. Aguirre, Corson L. Cramer, Vincent P. Torres, Thomas J. Hammann, Troy B. Holland, Kaka Ma, “Effects of the addition of boron nitride nanoplate on the fracture toughness, flexural strength, and Weibull Distribution of hydroxyapatite composites prepared by spark plasma sintering”, Journal of the Mechanical Behavior of Biomedical Materials, volume 93, May 2019, Pages 105-117

[10] Jay H.Lee, Joohyun Shin, Matthew J.Realff, *“*Machine learning: Overview of the recent progresses and implications for the process systems engineering field*”,* Computers & Chemical Engineering, Volume 114, 9 June 2018, Pages 111-121

[11] Marta Kolasa, Rafał Długosz, Tomasz Talaska, Witold Pedrycz. “Efficient methods of initializing neuron weights in self-organizing networks implemented in hardware”. Applied Mathematics and Computation Volume 319, 15 February 2018, Pages 31-47.

[12] Ali ihsan Sekertekin, Niyazi Arslan, Mehmet Bilgili, “Modeling Diurnal Land Surface Temperature on a Local Scale of an Arid Environment Using Artificial Neural Network (ANN) and Time Series of Landsat-8 Derived Spectral Indexes”, Journal of Atmospheric and Solar-Terrestrial Physics, Volume 206, 15 September 2020, 105328.

[13] Giulia Napolitano, Francesco Serinaldi, Linda See, "Impact of EMD decomposition and random initialisation of weights in ANN hindcasting of daily stream flow series: An empirical examination", Journal of Hydrology, Volume 406, Issues 3–4, 6 September 2011, Pages 199-214.

[14] Branimir Pavlic, Oskar Bera, Senka Vidovic, Lazar Ilic, Zoran Zekovic. "Extraction kinetics and ANN simulation of supercritical fluid extraction of sage herbal dust". The Journal of Supercritical Fluids, Volume 130, December 2017, Pages 327-336.

[15] Zlatan Car, Sandi Baressi Šegota, sNikola Anđelić, Ivan Lorencin and Vedran Mrzljak, "Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron", Volume 2020, Article ID 5714714.

[16] Piotr Wais, "Two and three-parameter Weibull distribution in available wind power analysis", Renewable Energy, Volume 103, April 2017, Pages 15-29.

[17] Bin Deng, Danyu Jiang, Jianghong Gong, "Is a three-parameter Weibull function really necessary for the characterization of the statistical variation of the strength of brittle ceramics?", Journal of the European Ceramic Society Volume 38, Issue 4, April 2018, Pages 2234-2242.

[18] Jos´e A. Montoya, Elo´ısa D´ıaz-Franc´es, Gudelia Figueroa P, "Estimation of the Reliability Parameter for Three-Parameter Weibull Models", Applied Mathematical Modelling, Volume 67, March 2019, Pages 621-633.

[19] Asghar Moeini , Kouroush Jenab, Mohsen Mohammadi, Mehdi Foumani, "Fitting the three-parameter Weibull distribution with Cross Entropy", Applied Mathematical Modelling 37 (2013) 6354–6363.

[20] Bryan Dodso , "The Weibull Analysis Handbook", Second Ed 2nd Edition.

CERTIFICATE OF COMPLETENESS

It is hereby certified that the dissertation submitted by NS Zeeshan Ali Cheema, Reg No. **00000172288**, Titled: ***Wind speed estimation using Artificial Neural Networks*** has been checked/reviewed and its contents are complete in all respects.

Supervisor's Name: **Dr. Imran Shafi**

Signature: ______________________

Date: 15-08-2020

```
clc
clear all
close all

%Input data for training%
input1=importdata('Temp Data;Press Data;Humidity Data)
targets1=importdata('Wind Speed Data')
testinput1=importdata('Test input Data)
testtarget1=importdata('Test Wind Speed Data')


targets=transpose((targets1));
test_input=transpose(testinput1)
test_target=transpose(testtarget1)

%Neural Network design
speed=newff(input,targets,[18,1],{'logsig','purelin'},'trainlm','learngdm','ma
e')

%speed=newff(input,targets,[5])

%Parameter selection
speed.divideparam.trainRatio=70/100;
speed.divideparam.valRatio=15/100;
speed.divideparam.testRatio=15/100;

speed.trainfcn= 'trainlm'
speed.inputWeights{1,1}.learnFcn='learngdm'

speed.layers{1}.initfcn='initwb'
%speed.layers{1}.initfcn='randnc'
%view(speed)

speed.trainparam.LearnRate=.1
%speed.trainparam.momentum=.86
speed.trainparam.perffunc='rmse'

speed.trainparam.epochs=600
speed.trainparam.goal=0
speed.trainparam.min_grad=10^(-15)
speed.trainparam.time=inf
speed.trainparam.max_fail=6
%speed.trainparam.mu=.01
speed.trainparam.mu_dec=.01
speed.trainparam.mu_inc=10
speed.trainparam.mu_max=10^(100)


%mem_reduc=1

%speed_trained =
train(speed,input,targets,'useParallel','yes','showResources','yes');

[speed,tr]=train(speed,input,targets)

plotperform(tr)


%validation=val(speed,input,targets)
```

```
%Results
output=sim(speed,test_input);
x=[1:1:length(output)];

plot(x,output,'r',x,test_target,'g')
legend('Predicted','Actual')
xlabel('Num')
ylabel('Wind Speed')
for i=1:1:length(x)
diff(i)=sqrt((output(i)-test_target(i))^2)/test_target(i);
rms(i)=sqrt(((output(i)-test_target(i))^2));
%accuracy(i)=sqrt((output(i)-targets(i))^2);
end
diff;
rms;
Error=mean(diff)
Rmse=(sum(rms))/(sqrt(length(x)))
mae=mean(rms)
xlabel('months')
ylabel('Wind Speed (m/s)')
%h2 = lillietest(log(targets),'Distribution','extreme value')
%wblplot(targets)
%plotwb(speed,'root',3)
%plotwb(speed)
%plotwb(net,'toLayers',2)
%plotwb(net,'fromLayers',1)
%plotwb(net,'toLayers',2,'fromInputs',1)
```