CHAPTER 1

INTRODUCTION

1.1 Background

The policy of India regarding the energy and power sector is driven by the present requirement and the future market demand. As the end of fossil fuel is very near almost within this century, so the government is aiming at two goal with one striker by focusing on renewable form of energy, which is much cleaner and environment friendly form of energy, to be utilized for production of electricity. The alternative form of renewable energy includes mainly wind, solar and tidal. Globally, the position of India in self-sufficiency of renewable power is 81st (*source : Ministry of new and renewable energy, MNRE*).

Globally, India is the third biggest consumer of primary energy after China and USA, having 5.6% share globally in 2017. The share of each of energy resource in India's energy consumption is as follows: from nuclear energy (8.7 Mtoe; 1.15%); renewable energy (21.8 Mtoe; 2.89%); Hydro electricity (30.7 Mtoe;4.07%); natural gas (46.6 Mtoe; 6.18%) ; crude oil (221.1 Mtoe; 29.34%) and finally coal (424 Mtoe; 56.26%), these all figures are from the year 2017. With reference to the year 2017, India's total imports was as follows : import of crude oil and its product - 198.8 million tons; LNG contribute 25.7 Mtoe ; Coal contribute 129.8 Mtoe, therefore the total of primary energy becomes 354.3 Mtoe which is around 47% of primary energy consumption. From the data it is observed that still in this 21st century, India mostly rely on fossil fuel having around 75% share. (*source : Ministry of new and renewable energy, MNRE*) Although our country India is electricity surplus country but still it is importer of electricity raw material due to its dependence on fossil fuel energy consumption and having generation system mostly running on fossil fuels. It has been estimated that by the year 2030, India's imports related to energy is going to be increased by 53% of total energy consumption, if India still rely on fossil fuel type energy consumption. It was a positive news for us in the 2015 year, when India stands among those countries who are power surplus having large production of power capacity.

It's a global fact that globally India ranked fourth in the wind power market, and still India's government is planning to add around 100,000 MW of capacity of solar power plant. It was estimated that in the calendar year of 2018, the investment in the energy sector by India was US$ 75 billion which was 4.1% of US$ 1.85 trillion global investment.

At present, the total installed power generation capacity in India is somewhat around and a bit above than 207.8 GW, of this total capacity, renewable is having a share of 25 GW, and in the renewable energy the majority is contributed by the wind energy.

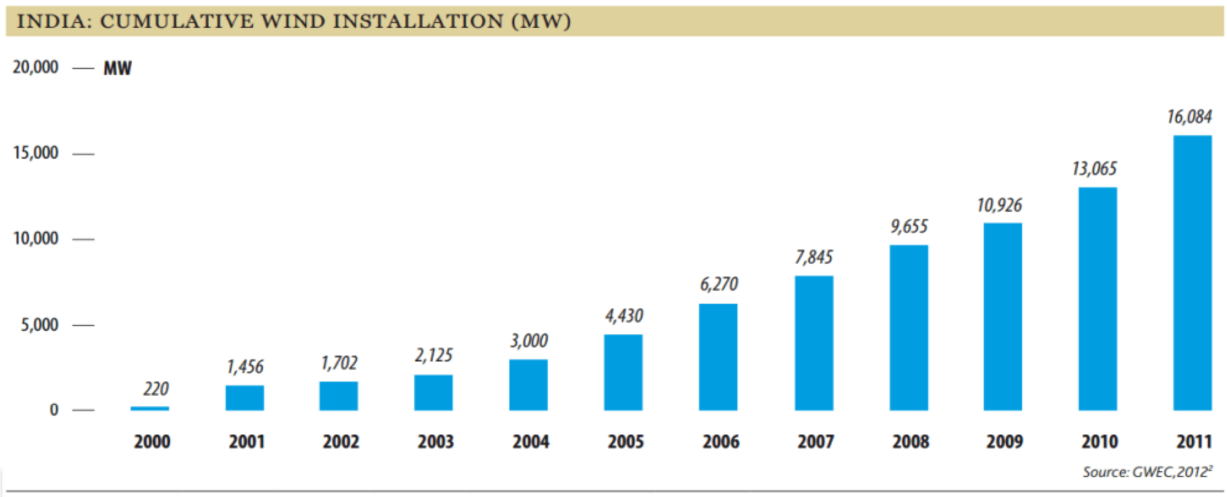

Figure 1.1. Wind installation (in MW) throughout India (year wise)

Currently, Across the world India is the 3rd vast wind energy market, and therefore this fact lays platform for excellent business chances for both foreign as well as the domestic players. The whole story start in the year 2011, when India saw the boom in this sector, when there was installation of more than 3 GW as new installation. Wind energy is now observed as a substitute and complimentary to the fossil and traditional fuel, which is much required for environment friendly condition. From Fig. 1.1 we can observe the Wind power plant installation (in MW) occurred throughout India.

While, tracking down our past, the planning done for the wind energy always hits the bull's eye. Whether we observe the 10th five year plan (2002-2007) or the 11th five year plan (2007-2012). As we can see that in the 10th five year plan, the target set was 1500 MW, while the installation done in actual was 5427 MW, therefore surpassing the set target. Similarly, in the 11th five year plan the set target was 9,000 MW, while the target achieved during the tenure was 10,260 MW. So, observing the past two five year performances, the government of India set the target for 12th five year plan was 15,000 MW normally, also at same time the inspirational bar was at 25,000 MW.

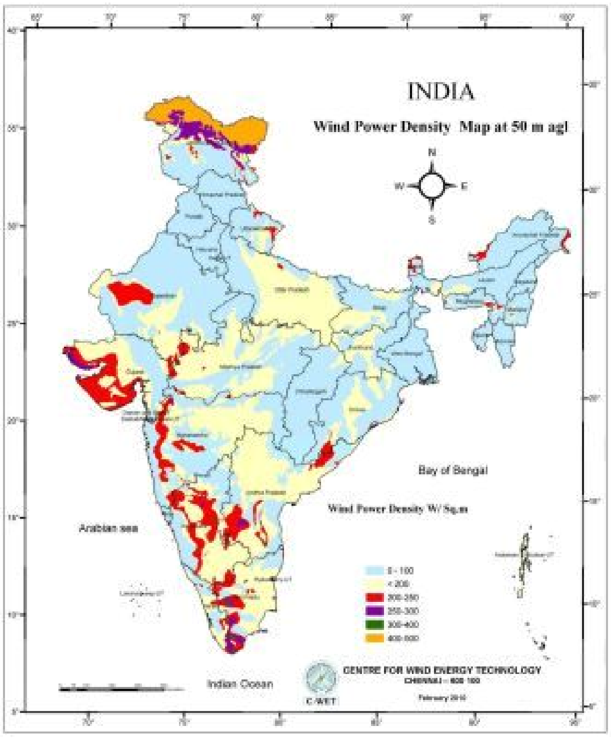

Figure 1.2. Wind power density and its potential across India

Wind power generation began in India for commercial purpose in the year 1986. The old models wind turbine were of lower capacity (<500kW) and hence there output is now falling short of the current market demand, also at the same time they are occupying the prime wind sites of India. The prime wind sites can be observed in Fig 1.2. So, in order to enhance the capacity of wind energy generation, the government has planned to restructure the old sites and replace the old low capacity wind turbines by the higher capacity and much efficient wind turbines. In the year 2011, the government of India through its ministry , which is Ministry of New & Renewable Energy has formularized the new set of rules for establishment of Wind energy projects. The main point of consideration in that guideline was that a site is viable for setting up of wind turbine plant only, if has a minimum wind density of around 200 W/m^2 at a hub height of 50 m.

India's Wind Capacity Crosses 10% Share In Overall Installed Base

In the past year that is 2018, there have been some fruitful results for India from the energy sector, as its renewable part achieve a major goal. The major boosting information was that

India's wind energy installed capacity had crossed the 10% share among the total installed base across India. Still, if we observe the energy sector keenly, then one can conclude that, at present also, wind energy dominates the renewable sector in India. If we make some study, then one can see that India's total installed capacity for renewable energy is 76 Gigawatts (as of 31st December 2018). While, in the last quarter of 2018, there was addition of 523 MW capacity to the total wind energy capacity. (*source : Ministry of new and renewable energy, MNRE*)

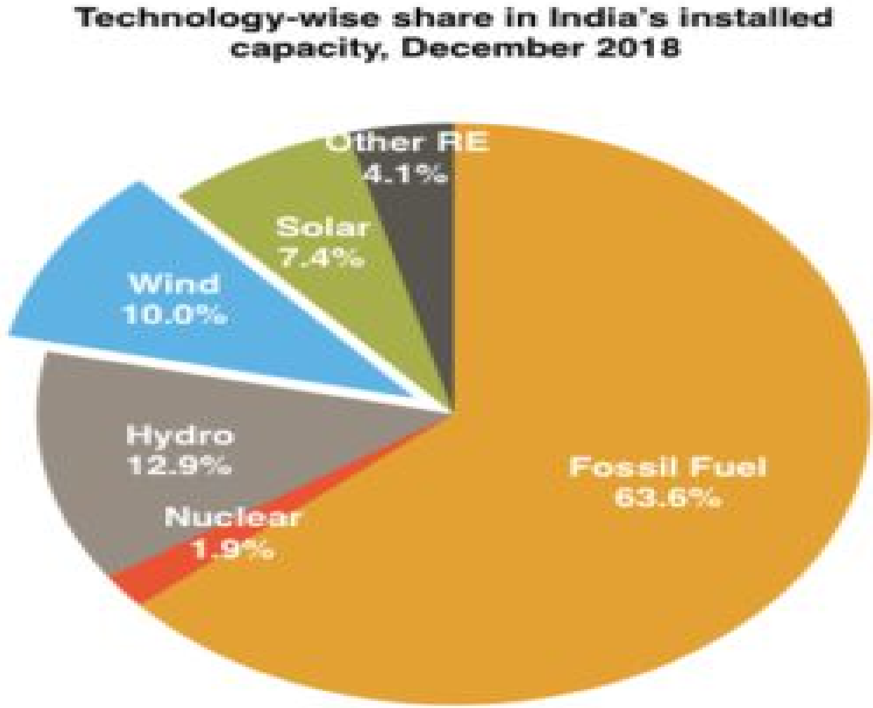

Figure 1.3. Percentage wise installed capacity of various renewable sector in India

In Fig 1.3 we can observe the percentage installed capacity of various renewable sector in India. During the past three to four years, one can observe in India that the capacity of wind energy system installed in India grows drastically from 23.4 Gigawatts in 2015, to 35.1 Gigawatts observed at the end of 2018 [43]. Around this period, the average capacity addition to the sector was 800 Megawatts which is quarterly addition. The current scenario of India's wind energy share is 4.6% of total installed capacity, in December 2018. In Fig. 1.4 we can observe the percentage contribution of total installed capacity in vast Indian state.

Currently India has the fourth highest wind installed capacity in the world with total installed capacity of 34.98 GW [43] as on October, 2018 against a target of 60 GW by 2022. Further, around 9.4 GW capacity is under implementation or have been tendered out [43]. National targets for offshore wind capacity additions of 5 GW by 2022 and 30 by 2030 declared.

STATE-WISE INSTALLED WIND CAPACITY INDIA

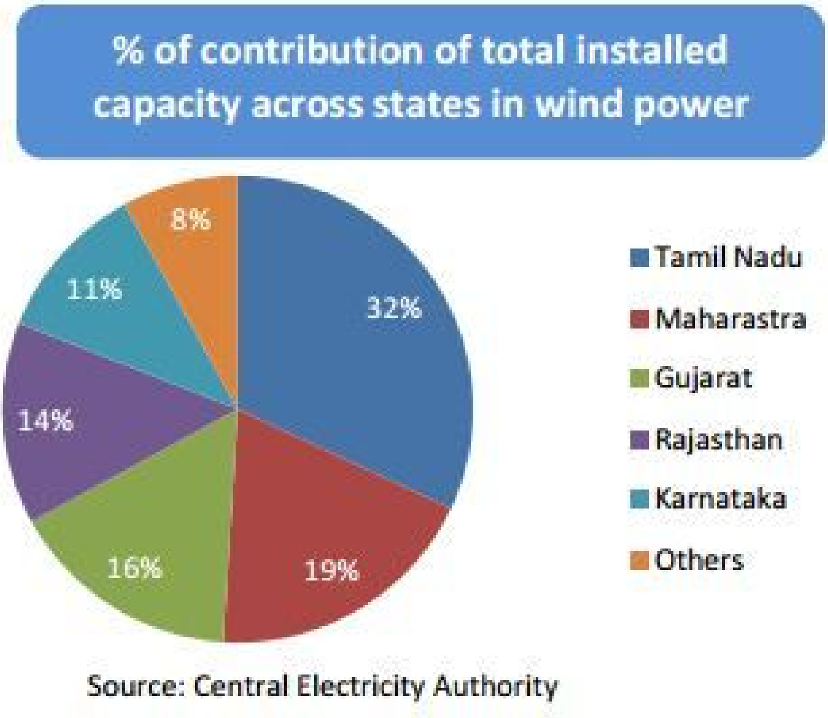

Figure 1.4. State wise percentage installed capacity of wind power

1.2 Forecasting of Wind Power System

Numerous nations over the world use wind power [1] as a sustainable power source asset. Exact prediction of power of wind maneuvers a huge task in producing constant power. Various elements are there influencing the anticipated intensity of breeze control forecast framework, similar to change in speed of the breeze, geological area, and condition of weather. Long term and short term prediction of breeze aka wind are the primary classifications of breeze control forecast systems. At the point when power expectation is completed before seconds, minutes or mid day, at that point it is known as short term prediction [2] of breeze speed or power. Then again, advance expectation or prediction of breeze power before a year, months or even few days is considered as long term wind derived forecast. For wind power estimating, short term or momentary based forecast of breeze power is viewed as progressively dependable. Those nations, which are wealthy in wind cultivates for the most part use distinctive wind influence forecast methodologies. Wind farm or ranch expectation systems are founded on either physical models or statistical models; additionally, cross breed algorithms which are a mix of different physical, statistical and factual models like : machine learning and deep learning are likewise well known. Determining of wind power utilizing physical techniques relies upon Numerical-Weather-Predictions [3] like humidity, temperature, snowfall, rain intensity, the smoothness of the surface, and many other such factors. If there should be an occurrence of statistical methodologies, just past information is taken into account without thinking about the weather conditions and physical factors. Statistical strategies use time-series and artificial intelligence [2-3] based methods to anticipate the breeze power or wind turbine power output. Hybrid or we can say, Half and half methodologies are a mix of statistical and physical strategies that endeavor both time-series arrangement and climate figure systems or climatic conditions.

Historically, many researchers coined various statistical methodologies related to prediction of wind power [2], energy and wind speed. Few statistical methodology include Adaptive-Neuro-Fuzzy-Interface, Auto-Regressive-Moving-Average (ARMA) [13], Radial Basis Function (RBF), Elman neural network, Neural Networks, adaptive wavelet neural network [4], Least-Square Support Vector Machine, deep learning [12], extreme machine learning etc. Similarly, considering breeze fields which are associated to the grid system of electricity, Landberg et al. proposed a wind power predictive system. In the same way, so as to predict breeze velocity and energy associated with the breeze. Cassola et al. utilizes the kalman filter output and he uses it as an analyzer. For predicting to be reliable, a researcher known as Liu et al. coined a scheme for wind power forecasting, in this method all inputs are decayed into bands of frequency and thereafter models of various kinds are trained as per the SVM model. Various types of kernels or small sub-groups are used while performing the training and testing of Liu's given methodology, also, relative Average-Relative-Error and Average-Squared-Error are performance checking tools for evaluation. Similar to the above, there was another proposed theory, Chitsaz et al.

inducts a very efficient system for prediction system used for predicting wind power which is made by using Wavelet-Neural-Network (WNN).

Comparative study of ARMA [7] (which is quite different from ANN[8]) and a technique known as Adaptive-Neuro-Fuzzy-Interface-System is done which was tested by Giorgi et al. for wind power prediction. Also, for optimizing energy from wind turbine and show improvement in power factor, a technique was suggested by Kusiak et al. Another methodology, was suggested by Amjady's et al for forecasting[31][32] wind power [9] using PSO [35][36] and ANN [8].

As it is very well known to the whole world that the traditional fuel or the fossil fuels are depleting at an alarming rate. So, in order to get substituted them by other much cleaner resources, other form of renewable energy must get tapped. The renewable sources include solar, wind, tidal etc. At present in India, the dominating energy is wind energy over the other type of renewable energy. Although the wind energy is impertinent and inconsistent as compared to other renewable sources, but yet it is a much fruitful and reliable way of renewable energy. Also, integration of wind power system with outdated grid system is also a problem for current engineers and researchers. For accessing and solving the problems as mentioned above, the forecasting tools of different types are used so that all problems can be overcome properly.

Now, In this work, various neural network algorithms are discussed and is used for predicting the wind power/ wind turbine power output, utilizing breeze velocity as the input parameter. The algorithms of neural networks are discussed in the form of case studies. Now, in a sequential way, The case study-1 is related to the comparison of three algorithms viz. NARX [5][10], NLIO[5] and RNN [6]. The second case is related to the comparative study of LSTM [21][22], RNN [20][22][29] and GBM/GBR [23]. While, the final case is related to comparative analysis of GA based SVM[24][25] with that of LR(Linear Regression), ANN/MLP(Multilevel Perceptron) [11][19] and RNN. Now, here the first two case studies basically used a hourly data as input and thus it predict the wind power in hourly form only. While, in the last or third case, the input was the hourly data, but the prediction is done on daily average basis. The dataset used for this work is the annual hourly dataset related to that of Calcutta Region, belongs to the year 2014-15. The parameters present in dataset are breeze speed and the wind turbine power output.

The algorithms are measured, compare and analyzed on certain performance parameters basis. These parameters are statistical[30] in nature. They are as follows: MSE [30], MAPE [30], RMSE, MAE [30], Variance. The mathematical or computational values of these parameters will decide that which of the following neural network algorithm performs well under the given process and condition.

1.3 Organization of Dissertation

The Dissertation is mainly classified into three case studies. Chapter 1 being the introduction to the topic and having a glance at India's power sector scenario. The Chapter 2 is the Literature Review related to the Thesis or Dissertation and Chapter 3 includes the Basic Statistical models used for forecasting of wind turbine power output. Then comes Chapter 4 which have our second case study describing about the comparative study of NARX, NLIO and RNN. In Chapter 5 we have our third case study discussing and elaborating the application of RNN, LSTM and GBM on our hourly dataset, further their performances are checked over Performance Parameters. In Chapter 6, there is an elaborative study of a novel hybrid algorithm called as Genetic Algorithm (GA) based Support Vector Machine (SVM), and further we have benchmarked it with other neural networks like : Linear Regression, ANN/MLP i.e. Multilayer Perceptron and RNN. Lastly, in Chapter 7 we have thorough conclusion of each case study and also suggested future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Forecasting [1] or prediction is a phenomenon , which can be explained as the anticipation of a few upcoming or future events , values by analyzing the historical or past information(data) along with present values. The field of prediction covers wide areas including industry ,business , environment , economics , etc. Forecasting or Prediction problems are classified into following types :

(i) Short term [2] Prediction (it includes time horizon of few months, weeks , days, hours, minutes, seconds)

(ii) Medium Term Prediction [34] (it includes time horizon for one year to two years)

(iii) Long Term [28] Prediction (it includes time horizon beyond 2 years)

In case of time series forecasting , there is involvement of data in time series form, which is expressed as a in-order series of observations belonging to a particular type of parameter. In this particular Thesis work the parameter to be predicted is power . While, the input parameter is wind velocity. The dataset used can be of many types depending upon number or types of variables. It could be uni-variate [17] or multi-variate. Uni-variate dataset includes only single type of data, while the Multi-variate [12] dataset have more than one parameters or variable in it. In the power sector or the Electricity market, the prediction helps the power supply companies to easily identify the trend of demands, which may occur in future. Thus, the forecasting basically help them to be ready for the upcoming situation. Also , the companies can become bullish or bearish with regards to electricity demands in the market , thus varying their price accordingly.

The methods for electrical power prediction are broadly classified as follows :

(i) Elementary Analysis

(ii)Procedural Analysis

(iii)Time Series Forecasting

Elementary Analysis is largely associated with venture analysis, seeking the long term goals of the power sector corporations. Procedural Analysis additionally uses the historical information/past data for locating out the long run demand. Moving Average could be a unremarkably used technique for Procedural Analysis. It is often thought as the unweighted average of the n-data points from the past values.

The above said technique is appropriate for short-term predictions. The 3rd and last technique is Time series forecasting , that essentially involves 2-forms of algorithms which are as follows:

(i) Linear Models

(ii)Non-Linear Models

Now as we know that the variety of Linear Models will be like : ARIMA [7], ARMA [13], AR [15], Smooth-Transition Auto- Regressive (STAR) [14] and it's different forms. In the above linear models, some predefined equations are used to slot down or fit in a mathematical algorithm for a uni-variate time series type problem. the chief drawbacks of all these methodologies are, they are not doing well for hidden dynamics and non-linearity prevailing within the information or data. Also, one major drawback is that the model acknowledged according to one series will never going to fit for the other type of sequence or series. The other type of models, which are Non-linear models engross methodologies like: ARCH [15], TAR[17]. Depending upon the nature of function appliance, a variety of deep learning algorithms are utilized. It includes CNN(Convolutional Neural Network) [22], Artificial Neural Network (ANN) [27] also known as multi layer perceptrons (MLP) [11] etc. These architectures or networks have been applied in varied fields like: natural language processing(NLP), image processing, software based computational statistical analysis or time series analysis etc. DNN algorithms (elaborated as Deep Neural Networks [18]) are able to recognize the hidden or buried designs(patterns) and underlying dynamics within the information or data provided to it, via self training process. In case of Wind turbine power , the information or data received is hourly dataset and it is mainly non-linear in nature. In order to decompose such systems of dynamical information we require modeling which can break down the underlying dynamics and hidden pattern of dataset. Deep learning [18] methodologies are competent enough for recognizing and using the interaction, association, pattern existing in the given information or data with the help of a self training or learning process.

In another way of classification of the models : In today's world there are many models or algorithms existing for wind power or wind speed forecasting. There is one such algorithm known as Numerical Weather Technique (NWT) [32][33] which not only just predict wind speed but it also predict the weather conditions. Basically, Forecasting or Predictive study is mainly classified broadly into following studies: statistical [30][33] approaches and physical approaches

[34]. NWT model fall under the second category which is physical methodology. Under NWT, the meteorological conditions are taken into account. Whereas in case of Statistical methodology the data considered is high on parametric values while the meteorological conditions are not taken into account. Models such as : Autoregressive (AR), autoregressive moving average model (ARMA), Moving Average and Autoregressive integrated moving average model (ARIMA), all such algorithms falls under statistical approach for predictive problem. The statistical methods such as auto-regressive conditional Heteroskedasticity (ARCH), auto-regressive integrated moving average model (ARIMA) or Box-Jenkins model, and smooth-transition auto-regressive (STAR) model are used for prediction of wind power but it could not adequately handle the noisy and nonlinear data. Thus, soft computing procedures, like fuzzy framework, genetic algorithm (GA) [26], and artificial neural network (ANN) were adopted for forecasting of wind power generation. ANN combined with other techniques like fuzzy or GA etc are to be among the one of most trusted techniques in the field of short-term wind power forecasting and showed improved accuracy over other techniques.

2.2 Work Analysis of Deep Neural Network using Adaptive Learning

Under machine learning algorithms, neural networks uses bias while performing the iterations during training and testing. These weights or bias are upgraded at the time of training as per the situation and hence these bias are changes adaptively. In case of DNN (Deep Neural Networks) [18], architectures are so much complex that it is difficult to optimize the parameters while performing the training process. So, to overcome this difficulty, many researchers utilizes the idea of Adaptive optimization [37] of Bias or weights in such a way that the process of training is performed by tuning (which is adaptive in nature) of the pre-trained DNN's . Certain examples associated to adaptive optimization can be mentioned here, Agostinelli et al. proposed an auto-encoder having stacked denoising feature along with adaptive multi-column technique for removal of noise from images getting corrupted. Another example is that of Kim et al., who proposed learning adaptive strategy, for improving the performance of pre-trained denoising auto-encoder. Basically, in this review ATL-DNN [37] technique utilizes the idea of adaptive training. So, in this hybrid algorithm, all the base learners are trained adaptively with time, and hence due to such feature this technique can be used for data which keeps on increasing with time.

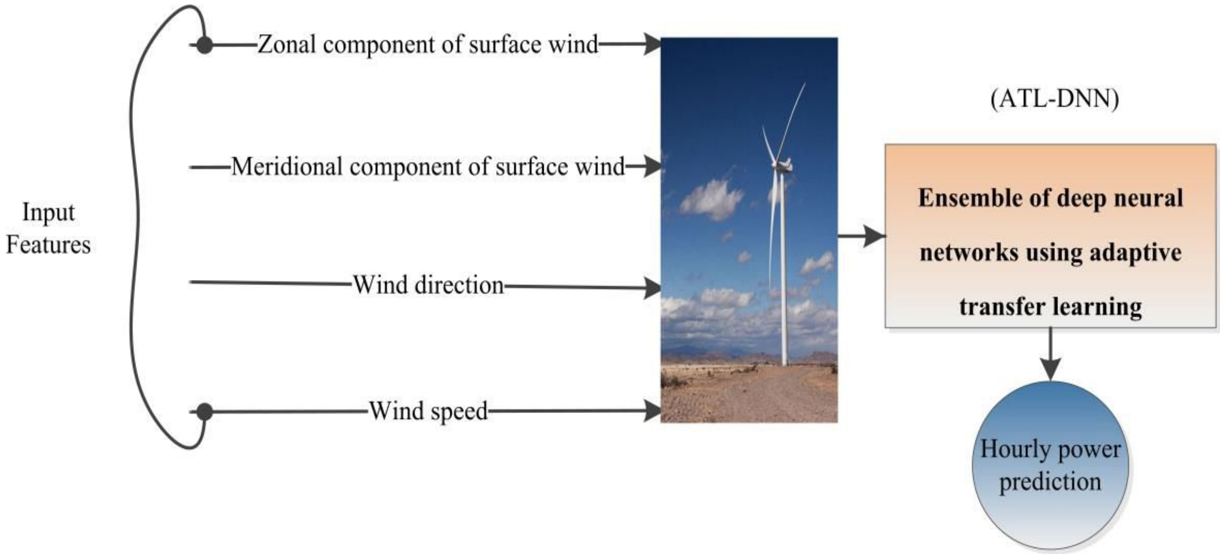

Figure. 2.1 Pictorial representation of ATL-DNN algorithm having input textures and predicted output

The served ATL-DNN algorithm is a hybrid regression technique in which training of auto-encoders are done at a periodic interval of 4 months adaptively. Fig. 2.1 displays basic concept of adaptive transfer learning (ATL) [37]. In order to explain this algorithm, the following process is done. First of all, out of several wind farms, the auto-encoder selects the data of wind farm on random basis at time of pre-training. Now, here we decide the source domain and the target domain, on basis of wind farm data performance in pre-training. source domain, become those data of wind farm on which base-learner is trained while other wind farms data out of several wind farm acts as target domain. Now, here there is cumulative effect of data, as the newly generated data also gets clumped up with the past data. This whole bunch of data is also utilized for fine-tuning of the base-learner and simultaneously generating adaptively new base-learner [37], which keeps on increasing with time. Fig 2.2 shows the use of adaptive transfer learning for the predictive purpose.

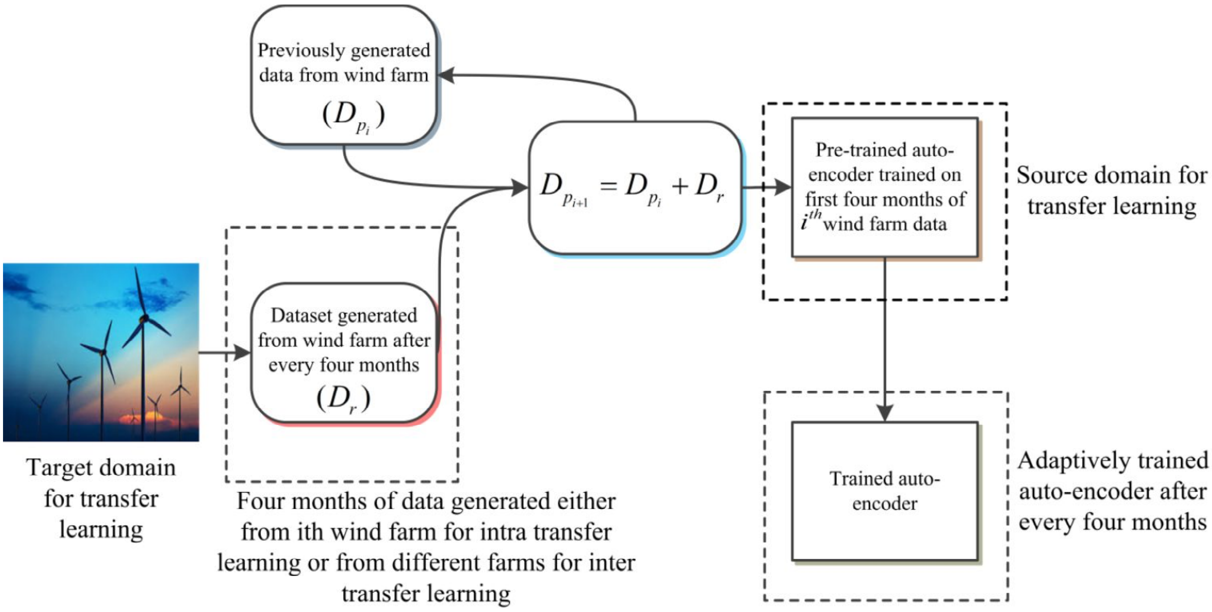

Figure. 2.2 wind power predictive system using adaptive TL

In the hybrid algorithm of ATL-DNN [37], base-learners can be trained adaptively after every four months. this algorithm works in two phases. During phase-1, all three auto-encoders are trained using the four months, eight months and twelve months of wind farm data. During the next phase that is the second phase, the meta-learner [37] is trained on the original features of wind farm data along with the predicted values of trained auto-encoders having data of months 13-16 . At the end, the testing is done with 17-20 months of data . The complete flowchart is shown in Figure. 2.3. In the flowchart we can observe that the dataset considered is of twenty months duration, taken directly from a wind farm. It is then consecutively divided into 4 months data, 8 months data, 12 months data, and then further 13 to 16 months of data are considered for auto- encoder. which means that the data from 13 to 16 months is directly fed to auto-encoder. Further, the 17 to 20 months data are taken for testing purpose. Further process can be observed from fig. 2.3.

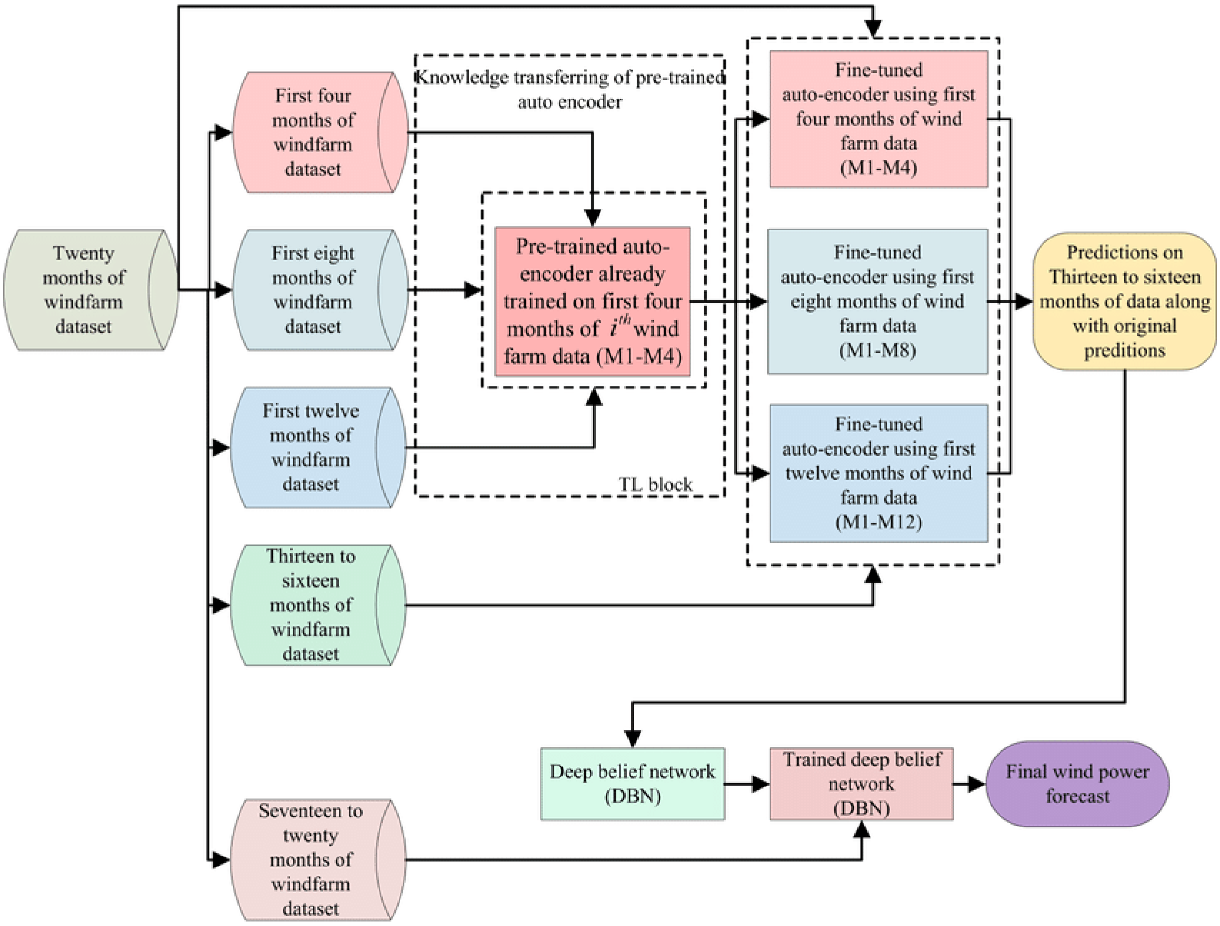

Figure. 2.3 Flowchart of the ATL-DNN algorithm.

The concept of Transfer Learning (TL)[37] is now coming in lime light very fast, because the labeling of dataset is time-consuming and costly as well. Also, at the same time TL is providing efficient weight or bias initialization. This Literature Review is basically discussing the idea of ATL i.e. Adaptive Transfer Learning to be applied in case of DNN [18] for wind power prediction. This hybrid algorithm of ATL-DNN [37] is used for short-term wind power predictions, where data is of form time series and is continuous and cumulative in nature. ATL i.e. Adaptive Transfer Learning brings a lot of advantage to the model. because, ATL brings excellent weight or bias initialization also it is assisting to use the continuous data which is generated on daily basis by several wind farms.

CHAPTER 3

BASIC STATISTICAL MODELS FOR TIME SERIES FORECASTING

3.1 Introduction

Most of us would have heard about the upcoming electricity market i.e. Renewable energy. Many of companies and market player would have invested in their coins in this sector because of its sustainability and environment friendly nature, but its main drawback is that it is much of environment dependent and also initial heavy investment. So, now the question arises is that such volatile sector is secure or not ? Now, How one can assure the investors that investing in this sector will make them healthy profits in upcoming days? Nobody could hit the bull's eye, but still we can produce an estimated value depending upon the past values and the upcoming demand related to this sector. one way to predict them is Time Series modeling. There are various statistical models which we can utilize in time series forecasting and use them as a tools in various analysis.

3.2 Understanding the problem statement and the dataset

Now on the table given is the problem related to time series guess of breeze turbine power output using wind speed which can be resolved over statistical tools and algorithms. Now, here we have considered the annual dataset of Kolkata region of year 2014-15 and it's an hourly dataset. Let's start working on the dataset by first observing its pattern over a given period of time and then also, observing it keenly, each of the parameters that is breeze speed and wind turbine output power over a given period of time. Firstly, we try to observe the pattern of the dataset. The graphical representation of the data pattern are as follows : Figure. 3.1 shows the complete data pattern for wind speed and wind power output, while Figure. 3.2 shows the wind speed pattern, and the Figure. 3.3 shows the wind turbine power output pattern. Figure 3.4. displays the division of data into training and testing sub groups.

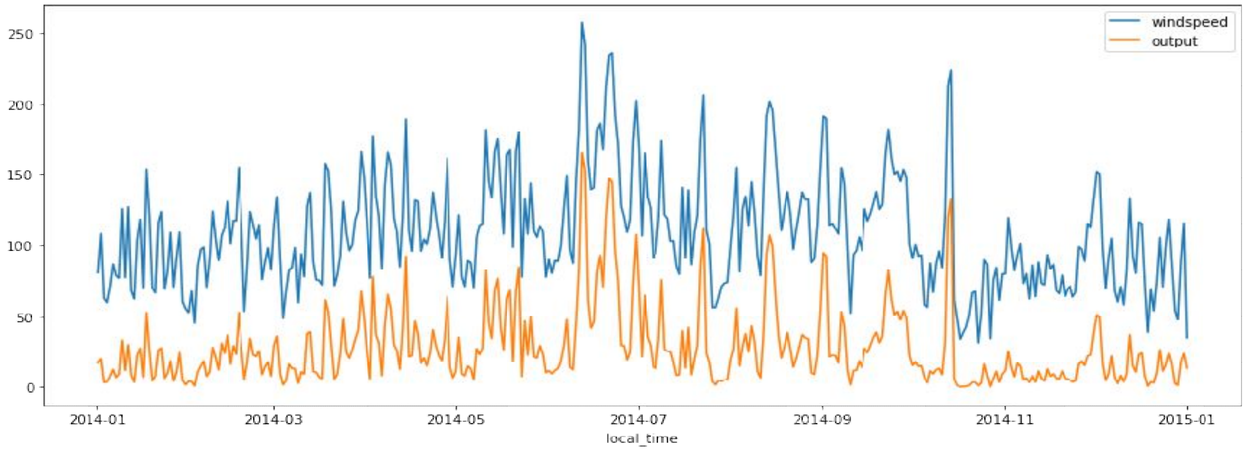

Figure. 3.1. Complete dataset pattern for the year 2014-15 i.e. wind speed and wind turbine power output

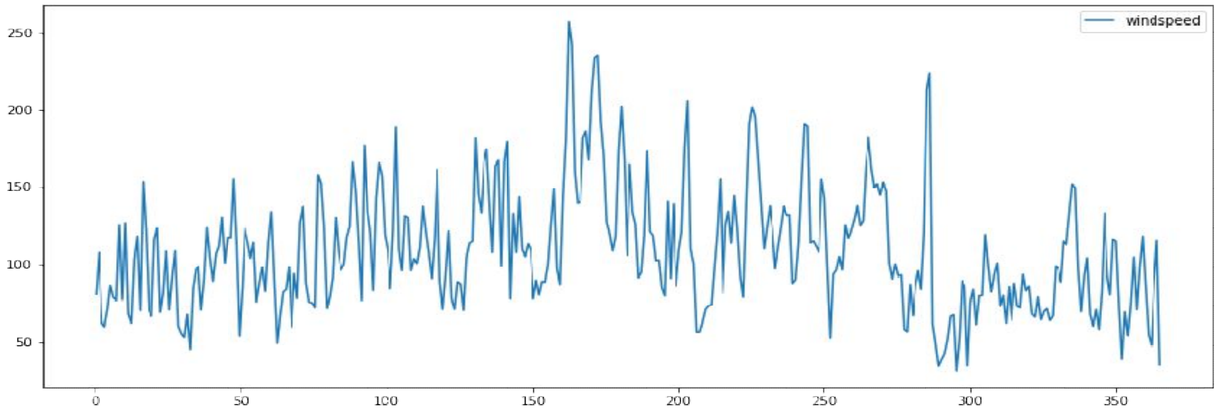

Figure 3.2. Dataset pattern for wind speed for the year 2014-15

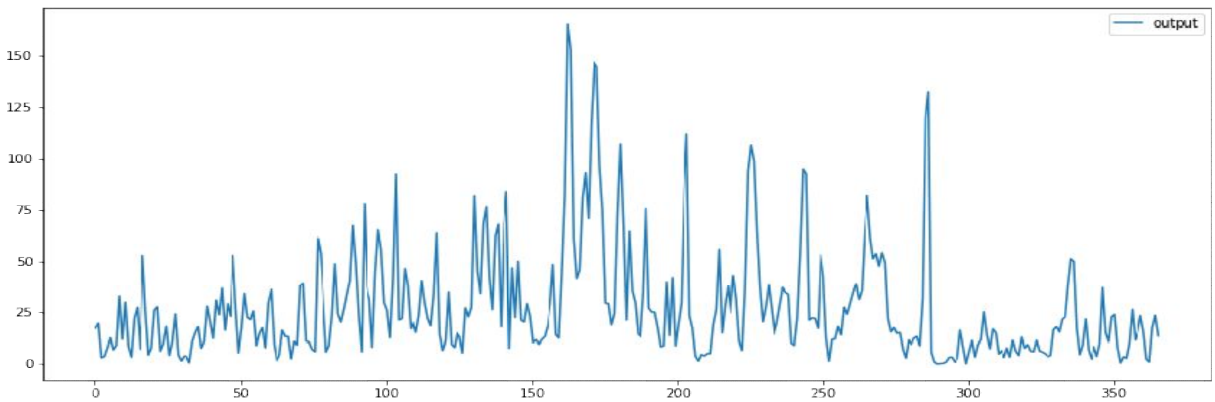

Figure 3.3. Dataset pattern for wind turbine power output for the year 2014-15

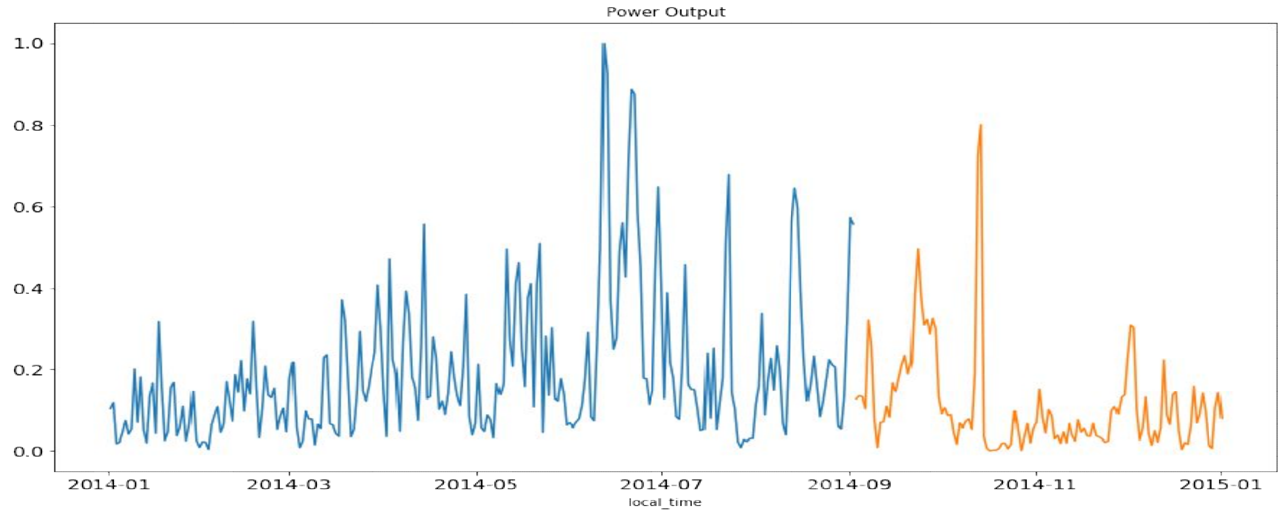

Figure. 3.4. Dataset pattern showing how it is divided between training and testing data

3.3 Installing Library (stats models)

The pre installed libraries and function which is used by me for performing Time series forecasting is statsmodels. The statsmodels library was installed in anaconda platform of python language, and then it was been called for, at various occasion in order to call certain statistical model in the source code.

3.4 Basic of Statistical models and their types

(a) Naive Approach / Naive Forecasting

Naive method [38] of forecasting is one of statistical [38] methodology in which the value at a particular instant of time is utilized to forecast for the same instant of time on next day. In this methodology, there are no adjustments for time dependent factors, which are causal in nature. This particular method is used for comparative analysis, with respect to other statistical methods. On many occasions, the researcher is provided with a kind of dataset, whose nature is quite stable across the time. So, if one wants to predict the wind turbine power output for next day, then just we have to consider the previous day value at particular instant, and then estimate same value for next day, at same instant of time. Such predictive methodology which presume that next predicted value is exactly same to the last value observed is called **Naive Model of Forecasting**.

Hence, the Naive approach can be represented by following equation

$$\hat{y}_{t+1} = y_t \tag{1}$$

Now, with the given dataset, when we apply the Naive approach on it using the libraries present in python then, the result which we obtain can be shown graphically in Fig. 3.5 as follows :

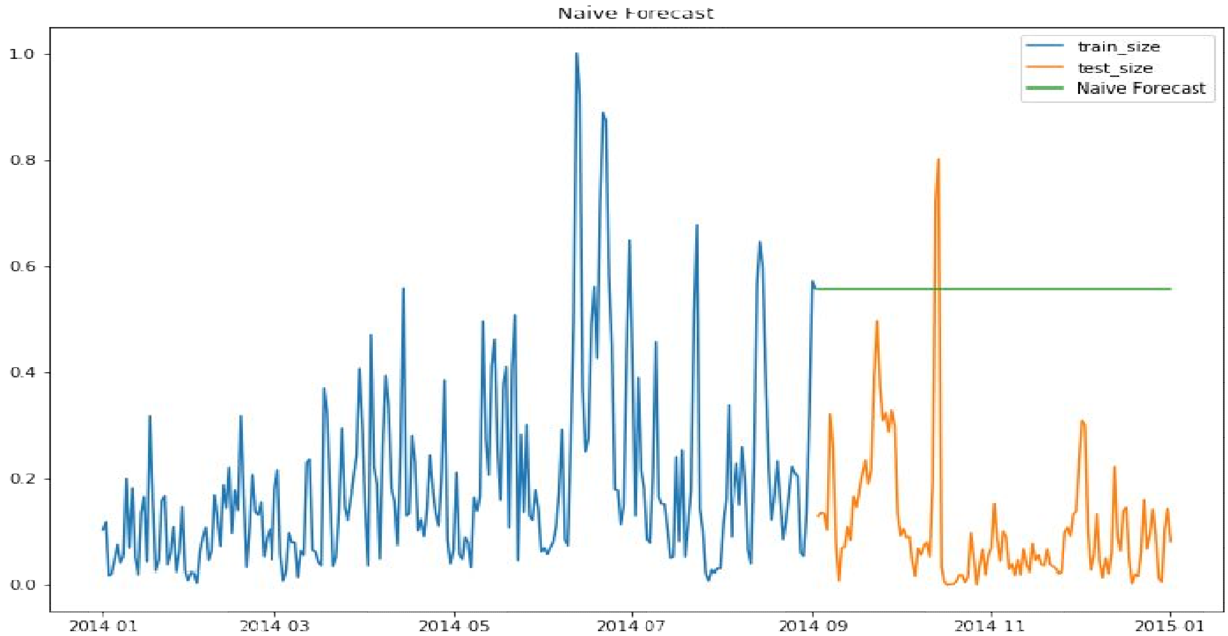

Figure. 3.5. Naive approach of forecasting with the given dataset

We will now calculate the performance parameters to find out the accuracy of our model on test data set.

MAE = 0.451790

MAPE = 26.173730

MSE = 0.214311

RMSE = 43.9164061439

Now, from the above values of the performance parameters and also, the graph shown, one can deduce that the Naive methodology isn't feasible for data depicting high variability. This model, only suits to dataset, whose pattern is quite stable within a given range.

(b) Simple Average Prediction Model

Simple Average [39] predictive model come into action when there is vast change in the parameter over a period of time, although the variation in the range of parameter is very thin or range is very minimal, and therefore the overall mean remain same or constant. Sometimes researcher came across the datasets having such pattern that there is lots of variation over period of time but due to low range variation, the mean remain constant, so in such pattern study, the parameter to be forecasted is basically the average of all the same values of the past day, or in a similar way we can say that for forecasting we consider daily average values of the past days. Such forecasting methodology which predicts the parameter to be predicted equal to that of mean of all past traced points is known as **Simple Average Forecasting Technique**.

Mathematically, it can be represented as follows from eqn. 2.

$$\hat{y}_{x+1} = \frac{1}{x}\sum_{i=1}^{x} y_i \tag{2}$$

Now, as said in this methodology, the researcher consider the past values, and then create the groups of that dataset either on daily or monthly basis but here the data is not considered on timely basis as in case of moving average [40], and then find out the mean of that value in order to predict the next day value of that same parameter. As one of the predictive methodology, this model work good at various situations.

Now, with the given dataset, when we apply the Simple Average [39] Forecasting technique on it using the libraries present in python then, the result which we obtain can be shown graphically as in Fig 3.6.

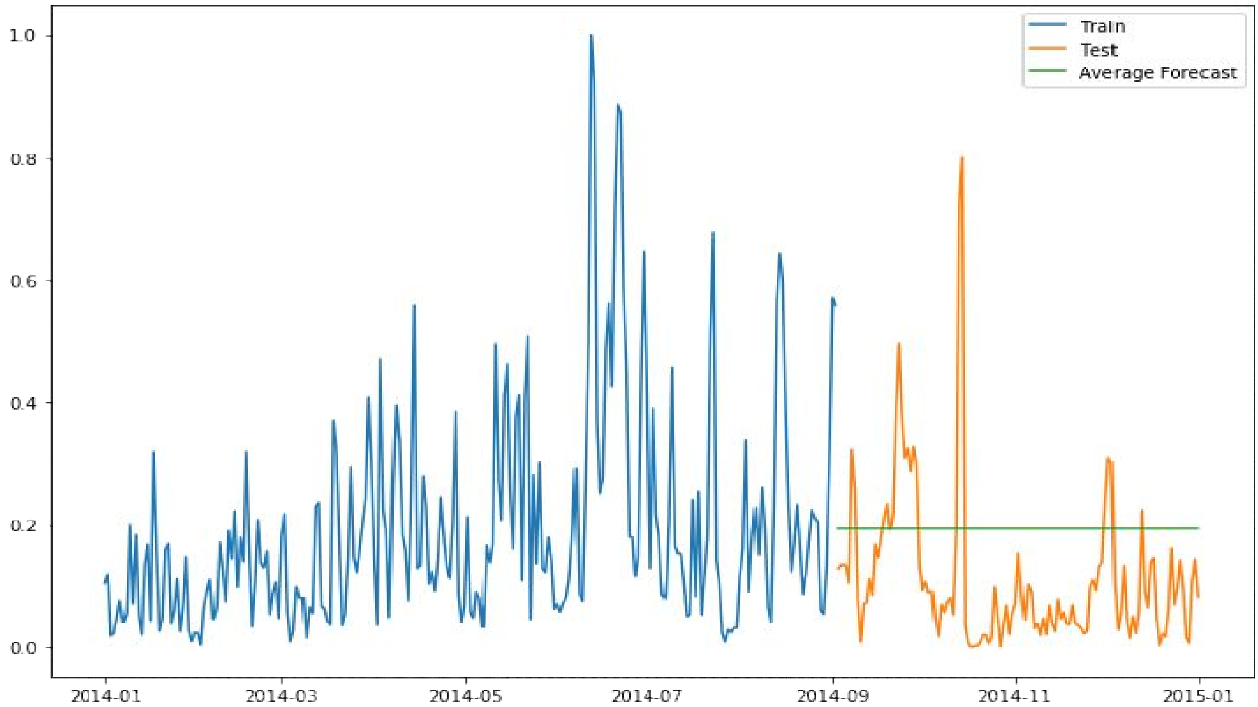

Figure. 3.6. Simple Average model of forecasting with the given dataset

We will now calculate the performance parameters to find out the correctness of our model on test data set.

MAE = 0.129991

MAPE = 8.563337

MSE = 0.022893

RMSE = 0.151304

Now, from the above values of the performance parameters, one can observe that this statistical model, enhance the performance very much as compared to previous model and hence we can deduce from above values that this methodology is at best when the mean of the dataset value remain same over the given period of time. Though, in this case Simple Average performs better than that of Naive method, but it doesn't mean that this concept can be generalized. Because on various other occasions Naive method outperforms the Simple Average predictive model.

(c) Moving Average Predictive Model

Sometimes researcher come across such dataset collection in which the parameters varies sharply and drastically over a period of time, and also having long ranges of parameter to be covered. So, in such case, the previous method of Simple Average can't work as the range of the parameter is quite high and the overall average doesn't remain constant over a period of time. So, in order to make improvement in the above methodology of simple average, there is a bit modification done. The changes made is that now, here the dataset is divided on basis of time periods and hence the mean of data is calculated to find out the mean value of parameter at that time instant. Therefore, we get the mean value at a particular instant of time. So, such methodology which uses the slots of time period in order to calculate mean of parameter for predicting out future values is called **Moving Average Model of Forecasting**. Therefore, for finding out the moving average [40] of the parameter, it basically involves , a time slot or sliding window denoted by size n.

So, by utilizing the moving average predictive model, researcher predicts the future values in a time series forecasting format, where the series is formed or created which is dependent on the mean finite number 'p' of past quantity. Therefore, for all i > p, the moving mean forecasting model can be mathematically shown as follows in eqn. 3

$$\hat{y}_i = \frac{1}{p}(y_{i-1} + y_{i-2} + y_{i-3} \dots \dots \dots . + y_{i-p} \qquad (3)$$

The effectiveness of moving average predictive model, can be varied and its performance can be enhanced by picking up the adequate and genuine values of 'p' for the series. Now, with the given dataset, when we apply the Moving Average forecasting model on it using the libraries present in python then, the result which we obtain can be shown graphically as in Figure.3.7.

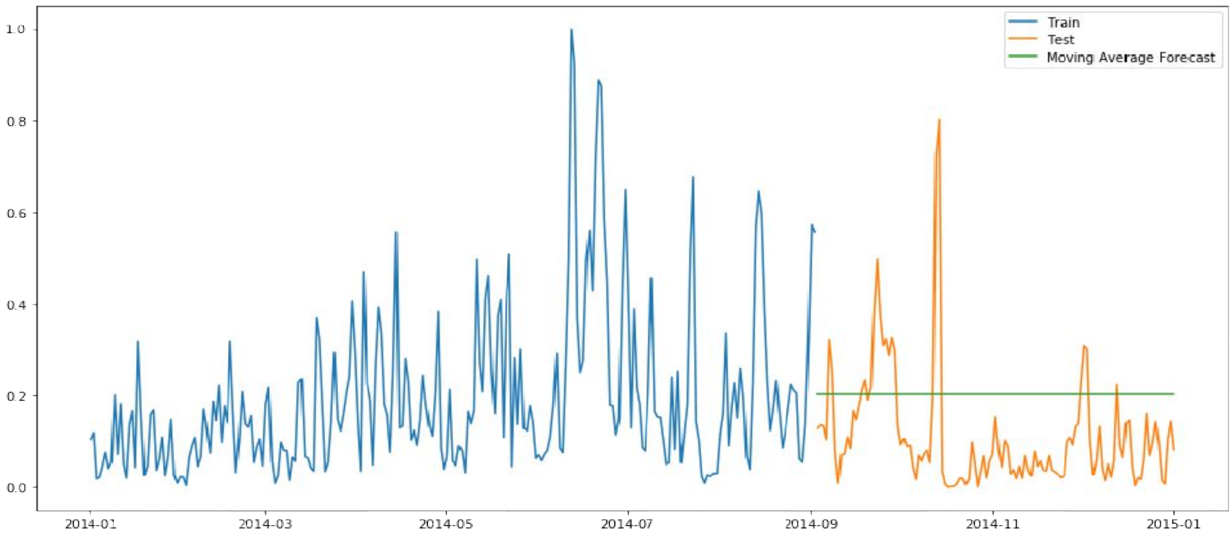

Figure. 3.7. Moving Average model of forecasting with the given dataset

We will find out performance parameters to audit the correctness of our algorithm.

MAE = 0.134951

MAPE = 8.907976

MSE = 0.024119

RMSE = 0.155303

Now, till now we can observe that the Simple Average predictive model, outperforms the Naive method and the Moving Average methodology for this particular dataset.

A slight modification in moving average method lead to an advanced version known as **Weighted or Biased Moving Mean** [40] method. As in the moving mean method, the past values influence the predicted values as per their weight or bias in the process, so this situation might cause variations in the results of performance parameters. Therefore, to overcome such situation in Weighted Moving Average method, there is consideration of weights of the past values in this case, therefore enhancing the presentation of the overall moving average algorithm, Such model is known as **Weighted Moving Average Predictive Model**.

So, a weighted or bias moving mean predictive model is a modified version of moving average, where within the time slot or sliding window, along the values, their weights or bias or impact factor is considered.

$$\hat{y}_i = \frac{1}{m}(w_1 * y_{i-1} + w_2 * y_{i-2} + w_3 * y_{i-3} \dots\dots\dots. + w_m * y_{i-m}) \quad (4)$$

So, here, along with selecting the size of sliding window, we consider the weight or bias of values (adding up to 1). Lets say, if researcher picks [0.40, 0.25, 0.20, 0.15] as weights, we would be giving 40%, 25%, 20% and 15% to the last 4 points respectively.

(d) Simple Exponential Smoothing Predictive Model

As we have come to know about the above methods namely, Simple Average [38] and Moving Average [39], we can observe that both models are performing in a very identical way, this is also proven by the values of performance parameters. So, to merge the features of the two method, the researcher uses the method known as **Simple Exponential Smoothing model** [41], where the higher weights are attached to the recent data, and lower weights are attached to the far or distant data, basically this method was a major improvement over Moving average, but still its performance varies from dataset to dataset. In this Simple Exponential Smoothing model, the prediction is done based on the values and their weighted averages. Here, the weights varies exponentially, as the observed values comes from past to recent. Because of the variation in the weights as one moves from past to recent. As said earlier, Higher weights are allotted to the recent value while lower weights allotted to the distant value, thus this exponential feature is introduced.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \dots\dots \quad (5)$$

here, $0 \leq \alpha \leq 1$ is the parameter of smoothing.

Also, unit-step-ahead prediction for time instant T+1 is the bias mean or weighted mean of all the considered values or observations for the given series $y_1, y_2, \dots\dots.., y_T$.

The parameter α. controls the rate at which the weights decreases exponentially.

After observing the scenario keenly, one will observe that the predicted outcome or the expected value $\hat{y}_x$ is the sum of two products: $\alpha \cdot y_t$ and $(1-\alpha) \cdot \hat{y}_{t-1}$. Therefore, another way of writing is as :

$$\hat{y}_{t+1|t} = \alpha * y_t + (1-\alpha) * \hat{y}_{t|t-1} \tag{6}$$

So here, basically in case of Simple Exponential Smoothing method [41] , we got two weighted moving average with different weights: α and 1−α.

So, from the above observation, it can be deduced that the previous expected value $\hat{y}_{x-1}$ is multiplied by 1−α., which therefore makes the overall expression recursive in nature. And this is the reason for calling out this method as **Exponential**. A weighted average between the most recent observation y_t and the most recent forecast $\hat{y}_{t|t-1}$ is equal to the prediction at time instant t+1. The simulation for Simple exponential smoothing model can be observe in Fig. 3.8

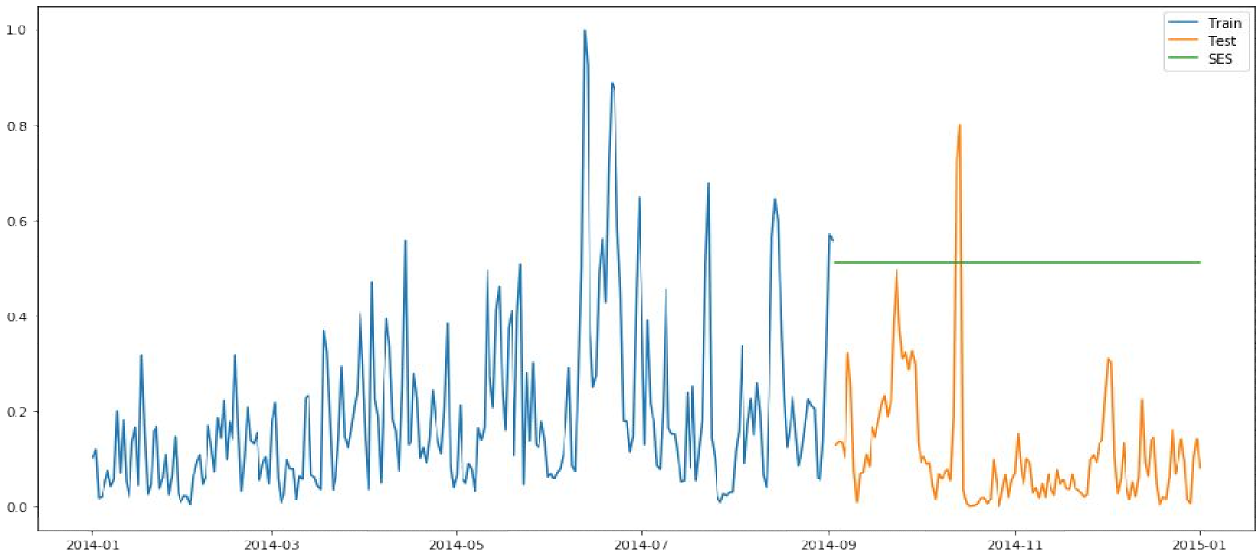

Figure. 3.8. Simple Exponential model of forecasting with the given dataset

We will now calculate performance parameters to check the accuracy of our model.

MAE = 0.406328

MAPE = 23.886077

MSE = 0.174679

RMSE = 0.417946

Finally, we observe that Simple Exponential predictive method having alpha as 0.6 value gives quite diverting results from that of simple average and moving average model. Still we can further tune this model for getting improvement in this statistical model.

(e) Holt's Linear Trend Method Predictive Model

After performing prediction with four of the statistical predictive models, with certain assumptions, now there is a need of a model in which the researcher could predict the trend precisely without any need of any assumptions. So, such one of method who do not account any assumption in its modeling is known as **Holt's Linear Trend method** [42].

Now, In this methodology, the dataset is considered and then the dataset is resolved or broken into certain components Like: Residual, Seasonality and Trend. Any dataset can be used for Holt's Linear Trend method because each dataset have different nature and trend, whether increasing or decreasing, so in either case, it is suitable for use.

Basically, the idea behind this Holt's Linear Trend model was to implement the simple exponential smoothing model with the trend, which means that the simple exponential predictive model can be applied at both the things which are Level (the mean value of the data in the series) and Trend. In order to show it mathematically, we have shown it with the help of three eqn. : first is for forecast; second one for level equation; third one represent trend equation. basically, the level eqn. and the trend eqn. are combined to get the expression for the forecast.

Forecast Equation : $\hat{y}_{t+h|t} = \ell_t + hb_t$ (7)

Level Equation : $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ (8)

Trend Equation : $b_t = \beta * (\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$ (9)

The predicted values of the models are called Level.

The eqn. pertaining to trend clearly screen us that it is a bias mean of the probable trend at time t based on ℓ(t)−ℓ(t−1) and b(t−1), which is the past guesses of the trend.

Now, further one will be adding all these eqn.(8) and eqn.(9) to formulate Forecast eqn. One can also formulate a multiplicative predictive eqn. by multiplying trend eqn. and level eqn. as an alternative of adding it. Now, if there is a variation in trend which is linear i.e. trend decreases or increases linearly, then we use additive eqn. but if there is variation in trend which is exponential i.e. trend decreases or increases exponentially, so we use multiplicative eqn. From the past experiences it has been observed that multiplicative eqn. is more stable or static predictor, whereas additive eqn. is simple to understand. The simulation for Holt's Linear Trend Model can be observed in Fig. 3.9.

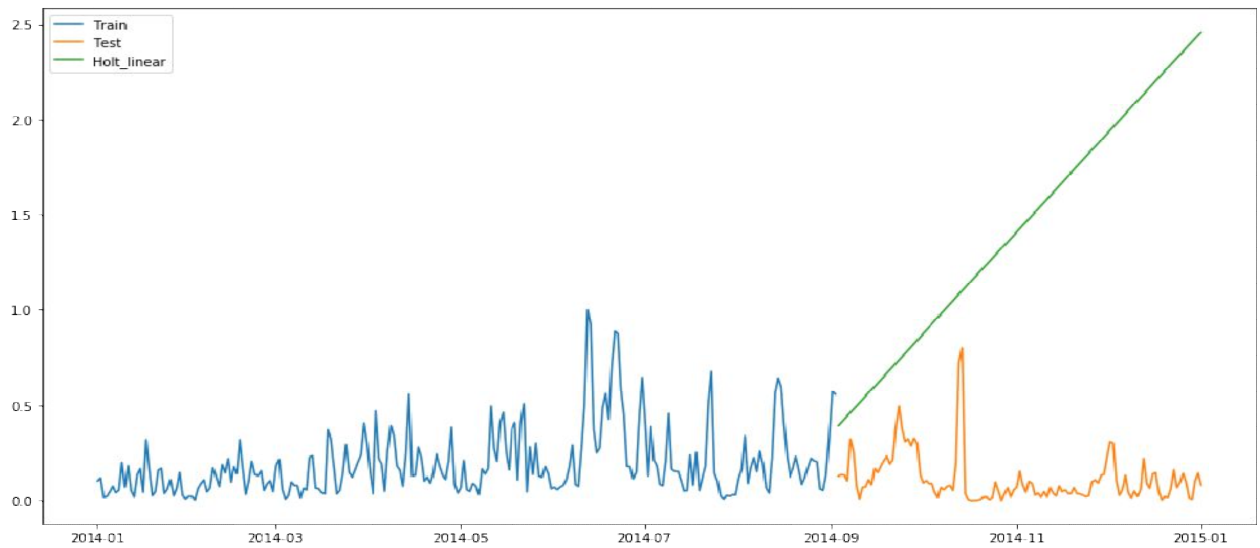

Figure. 3.9. Holt's Linear Trend model of forecasting with the given dataset

We will now calculate performance parameters to check the accuracy of our model.

MAE = 1.314510

MAPE = 69.010254

MSE = 2.151377

RMSE = 1.466757

We can see from the above performance parameters values that this Holt's Linear Trend Predictive model does not perform very well. But, still we cannot generalize this theory, as it has been observed previously on many occasions that this Holt's Linear Trend algorithm performs better than that of other statistical models. But with this dataset, this model have performed least, as can be observed from the comparative table, given later.

(f) Holt's Winter Method Predictive Model

First of all, in order to get the concept more clear for this particular method, one should have known the term "Seasonality". The term Seasonality is basically related to the frequency of occurrence of any of instant. For example. In the business of a hotel, situated in a hill station, experiences a heavy rush during summer season and there is a damp period during the rest of other season like : winter season, so from this one can conclude that the hotel will be getting maximum profit during summer season. And also this trend will be repeating itself every year. This repetition of whole cycle every year is known as "Seasonality". So, in the field of data science, the dataset which are showing repetitive nature after certain interval of time is said to be suffering from Seasonality.

All the models of forecasting discussed previously doesn't consider the factor of seasonality, which is a major factor impacting the dataset, while prediction. Therefore, there is an urgent need of a method which took into its account the Seasonality factor, along with the consideration of trend (which was considered in earlier models), for predicting the wind turbine power output. One such model which considers both Seasonality and Trend is Holt's Winter Method. The main concept of this model is to apply exponential smoothing to the seasonal components along with trend and level.

So, to include Seasonality factor into consideration Holt's winter method is among the best model to be used. The Holt's Winter model, took into account one equation regarding seasonality along with the other three eqn. which are trend, forecast and level. They are elaborated mathematically as follows: which are : one for trend b_t , one for the level ℓ_t, and one for the seasonal component denoted by s_t, with smoothing parameters α, β and γ.

Level $$L_t = \alpha(y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1}) \tag{10}$$

Trend $b_t = \beta(L_t - L_{t-1}) + (1 - \beta)b_{t-1}$ (11)

Seasonal $S_t = \gamma(y_t - L_t) + (1 - \gamma)S_{t-s}$ (12)

Forecast $F_{t+k} = L_t + kb_t + S_{t+k-s}$ (13)

where s is the length of the seasonal cycle, for $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$.

Now, from above equations we can observe that the Level eqn. depicts a weighted or bias mean between the non seasonal forecast for time t and the seasonally adjusted observed values. The Seasonal eqn. shows a weighted or bias mean between the seasonal index of last year and the seasonal index of the current year. The Trend eqn. is same as Holt's Linear Method.

In the Holt's Winter predictive model, we can utilizes both technique viz. multiplicative as well as additive. The multiplicative method is used whenever the changes in level of series are proportional to that of the seasonal variations. While the additive method is preferred when this seasonal variations are constant or remain same throughout the series. The simulation for Holt's Winter model [42] can be observed in Fig 3.10.

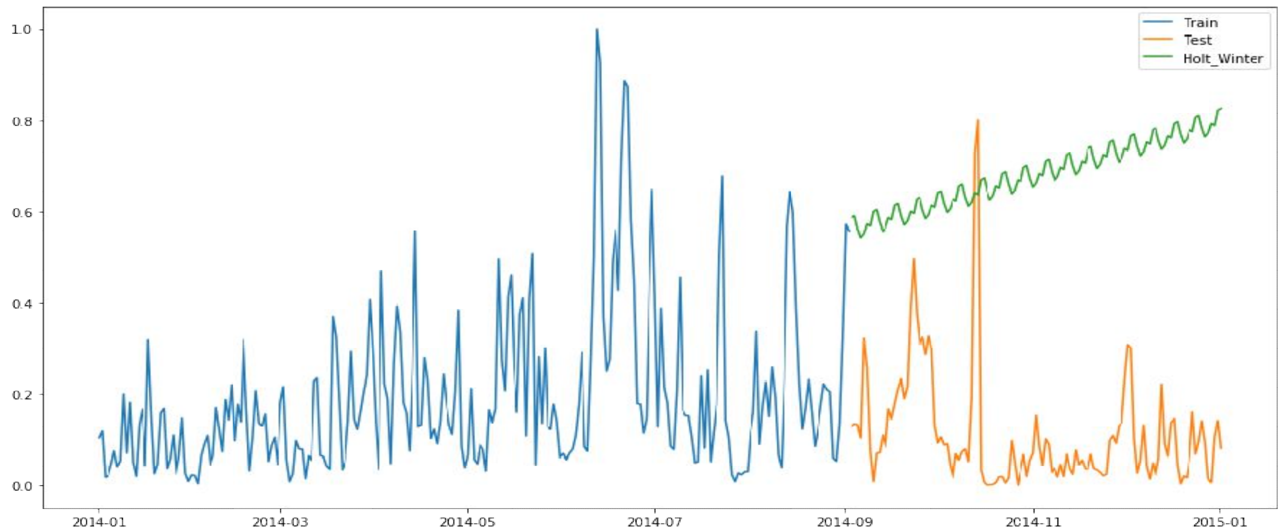

Figure. 3.10. Holt's Winter method model of forecasting with the given dataset

We will now calculate performance parameters to check the accuracy of our model.

MAE = 0.572109

MAPE = 32.061091

MSE = 0.349263

RMSE = 0.590985

We can observe from the above graphical simulation that this method or predictive model is not performing well as per its standard. This is because the dataset is lacking the seasonality factor in its pattern, therefore leading this method to underperform. However, This model is of uttermost importance whenever the dataset is used having some seasonality factor in it. In this particular study, I have chosen default parametric values while restructuring this model.

(g) Seasonal ARIMA (SARIMA) Predictive Model

Autoregressive Integrated Moving Average (ARIMA) [7] is a very prominent model for predictive analysis and is a model always considered by data scientists. The specialty of this model is that it basically correlates the data among each other. An improved model of ARIMA [7] is Seasonal ARIMA [7], which took into account the Seasonal factor also just like Holt's Winter method, along with correlating of the data which is a major feature of ARIMA [7].

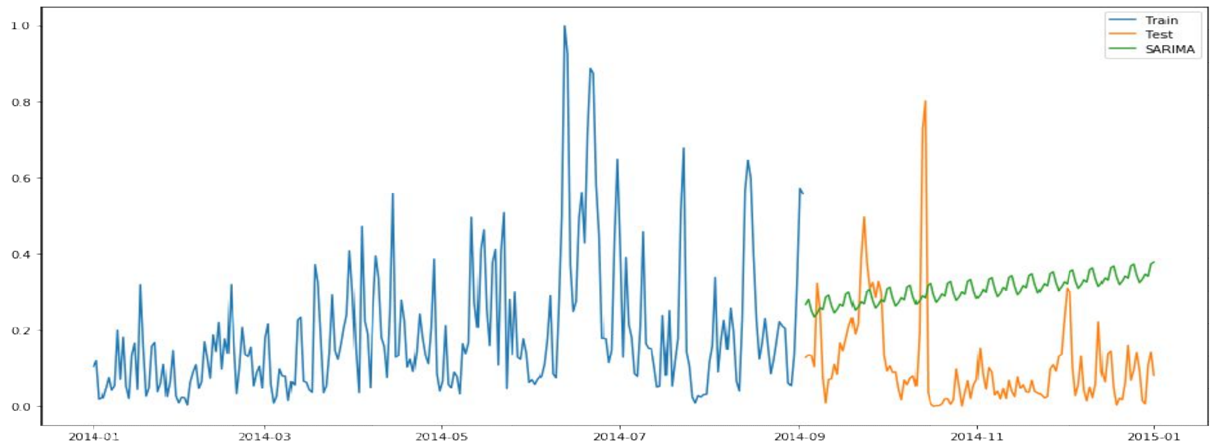

Figure. 3.11. Seasonal ARIMA (SARIMA) model of forecasting with the given dataset

The simulation for SARIMA can be observed in Fig. 3.11. We will now calculate performance parameters to check the accuracy of our model.

MAE = 0.220475

MAPE = 13.883405

MSE = 0.057078

RMSE = 0.238910

From above simulation one can deduce that using Seasonal ARIMA (SARIMA) gives out a solution which is quite close to that of Simple Average model and Moving Average model. But the MAPE for SARIMA is much Higher than the Simple Average and Moving Average this shows that SARIMA underperforms as compare to that of Simple Average and Moving Average model.

Now, comparing the models on basis of performance parameters.

TABLE 1. Comparative study of Statistical models

PARAMETERS	NAIVE MODEL	SIMPLE AVERAGE MODEL	MOVING AVERAGE MODEL	SIMPLE EXPONENTIAL SMOOTHING MODEL	HOLT's LINEAR TREND MODEL	HOLT's WINTER MODEL	SARIMA
MAE	0.451790	0.129991	0.134951	0.406328	1.31451	0.572109	0.220475
MAPE	26.17373	8.563337	8.907976	23.886077	69.01025	32.06109	13.883405
MSE	0.214311	0.022893	0.024119	0.174679	2.151377	0.349263	0.057078
RMSE	0.462938	0.151304	0.155303	0.417946	1.466757	0.590985	0.238910

From the above table we can deduce following:

On basis of MAPE, MAE, MSE, RMSE Simple Average model performs better than the other, while Holt's Linear trend model performs worst of all. Also, this analysis might change a bit with variation in the data pattern.

CHAPTER 4

PREDICTIVE ANALYSIS USING NARX, NLIO AND RNN

4.1 Introduction

The power utilities now a day's focusing on the use of renewable energy at massive level due to increasing awareness about environment and depleting natural resources. Wind power is one of the main renewable form of energy but due to the intermittent nature of wind speed, it becomes important to have a precise prediction of speed of wind and wind turbine power before it can be used as primary source of electricity. In this paper three artificial intelligence methods NARX[5-10], NLIO[5] and RNN[6] Networks are used for short-term wind power forecasting using the data of Kolkata region of India.The simulation results suggest that RNN [6] is able to forecast the wind power better than NARX [5] and NLIO [5] network.

To use the wind generation as a regular power supply, it is important to forecast the wind power generation more accurately with minimum possible error. The yield of a wind energy system generator is influenced by meteorological state of affairs as the power generated by the wind turbine is dependent upon various metrological parameters such as velocity of wind and density of air is shown by eq. 1 below.

$$P_{wind} = (1/2)\,\rho\, A v^3 C_p \tag{14}$$

where,

P_{wind} is the output power of a wind turbine (kW)

A is the wind turbine Rotor area (m^2)

ρ is the air density (kg/m^3)

v is the velocity of wind (m/sec)

C_p is the utilization factor of wind energy

As from eq. 14, since all other parameters such as rotor of wind turbine, air density etc are constant for a given conditions, wind speed is the major factor influencing the wind power generation. So in order to forecast the wind power, the wind data which is recorded on hourly basis is considered as an important parameter.

The statistical methods such as auto-regressive conditional Heteroskedasticity (ARCH), auto-regressive integrated moving average model (ARIMA)[7] or Box-Jenkins model, and smooth-transition auto-regressive (STAR) model are used for prediction of wind power but it could not adequately handle the noisy and nonlinear data. Thus, soft computing procedures, like fuzzy framework, genetic algorithm (GA), and artificial neural network (ANN) or Multilayer perceptron (MLP)[11] were adopted for forecasting of wind power generation. ANN combined with other techniques like fuzzy or GA etc are to be among the one of most trusted techniques in the field of short-term wind power forecasting and showed improved accuracy over other techniques.

In this paper three different forecasting tool such as nonlinear autoregressive network with exogenous inputs (NARX), nonlinear input-output (NLIO) network and recurrent neural networks (RNN) are used to predict the day-ahead wind power generation. The predictive flowchart diagram for short-term wind power forecasting is shown in fig. 4.1 below.

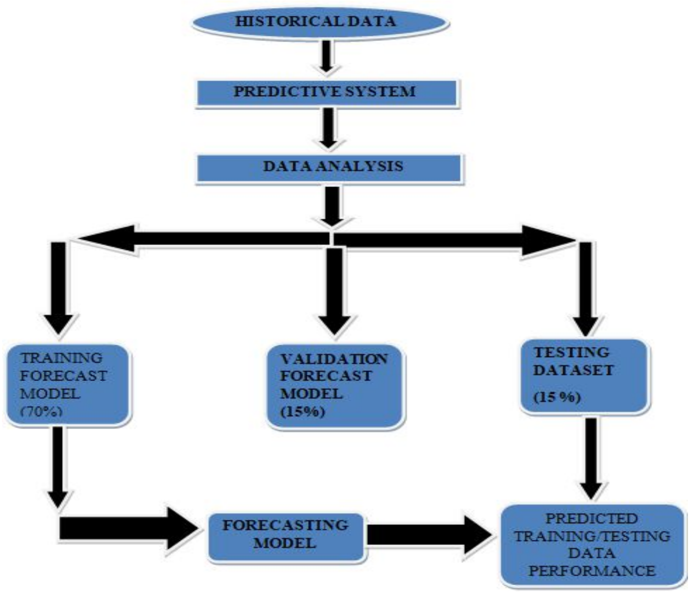

Figure. 4.1. Flowchart diagram used for Forecasting

The variety of undesirable attributes like : temperature , pressure , altitude etc which are of high clamor and non-stationary in nature could not yield very accurate results.

Hence the wind speed is taken as input to the proposed neural network as it is simplest way to predict the short-term wind generation. The data of Kolkata region in India has been used for wind power forecasting. The data considered in this paper is the hourly data of 2014 (1st Jan to 31st December) and 2015 (1st Jan to 31st December). This data set was split into the 24-hours data each, thus grouping them for 365 days. The 2014 dataset was used for training and the 2015 dataset was used for testing. The simulation results shows that RNN is the better candidate than NARX and NLIO for short-term wind power forecasting. Although it's a proven fact that RNN perform better than the feedforward network, but ,as the neural networks are mostly trained and tested for multiple input variable, it is in this research paper the neural networks performance are compared for single input variable , with the help of several parameters checked at the same time. Here the research work is related to the single input that is wind speed , which is used for wind power prediction.

4.2 Algorithms Description

Following three algorithms are used in this work as described below, with following features:

a) The size of hidden layer for each network is 10 .

b) The average CPU time for NARX is 2 sec per iteration , 3 sec per iteration for NLIO and less than 1sec per iteration for RNN.

c) In all the algorithms, Wind velocity (in m/sec) is Input parameter , while the targeted parameter or the output parameter to be predicted is Output Power (in kW).

4.2.1 Nonlinear Auto-Regressive Exogenous (NARX):

NARX has exogenous sources of input, thus it establish relation to the present approximation or estimation of the given period arrangement for both previous and past values of same series. Also, this model holds an error factor which relates in the way that information pertaining outside the system will affect the current inference of the time series arrangement , which is to be estimated specifically. Thus it implies the model relates to the present estimation of the given period arrangement for both:

a) previous/past values of similar sequence or series.

b) present and previous values of the driving (exogenous) series — which means, of the superficially determined series that influence the series of interest.

Mathematically , NARX can be represented as follows :

$$yt = F(yt\text{-}1 , yt\text{-}2 , yt\text{-}3 ,....... , ut , ut\text{-}1 , ut\text{-}2 , ut\text{-}3 ....) + \varepsilon_t \tag{15}$$

in equation (15) ,

y is defined as - variable of interest

u is defined as - externally determined parameter.

In this NARX equation , knowledge about *u* assists in predicting *y*, as do previous values of *y* itself. Here *ε* is the "error factor" (many a times called as noise). function F is known as Non Linear Function , it can be a sigmoid network or a wavelet network or a neural network. (NARX) is such a type of Dynamic Recurrent Network that along with feedback connections, it is also enclosing several layers of the network. The NARX network structure includes - tapped delay lines and 2-layer feedforward network, with a transfer function of sigmoid in the hidden layer and a transfer function with linearity(linear function) in the output layer. In NARX network , we basically employ the past values , in order to forecast the future values of output .The NARX network can be pictorially depicted as shown in fig.4.2. This implies that the NARX network uses the concept of Feedback or Error signal as follows:

consider a scalar(or vector) with time t and output W ;

{ $x(t_0)$, $x(t_1)$, ... , $x(t_{i-1})$, $x(t_i)$, $x(t_{i+1})$,...}

t is real value , while x(t) is a continuous signal.

The discrete signal can be configured out by constant sampling as shown in equation below , where sampling period Δt is introduced according to the Nyquist sampling theorem.

$\{x[t]\} = \{ x(0) , x(\Delta t) , x(2\Delta t) , x(3\Delta t) ,....\}$

Estimating x at some future time

$x[t+s] = f(x[t] , x[t-1] ,x[t-2], ...)$

here , s is known as - Horizon of Prediction .

if s=1 ; it implies, *Uni-Step* Ahead Prediction

else, it is called *Multi-Step* Ahead Prediction

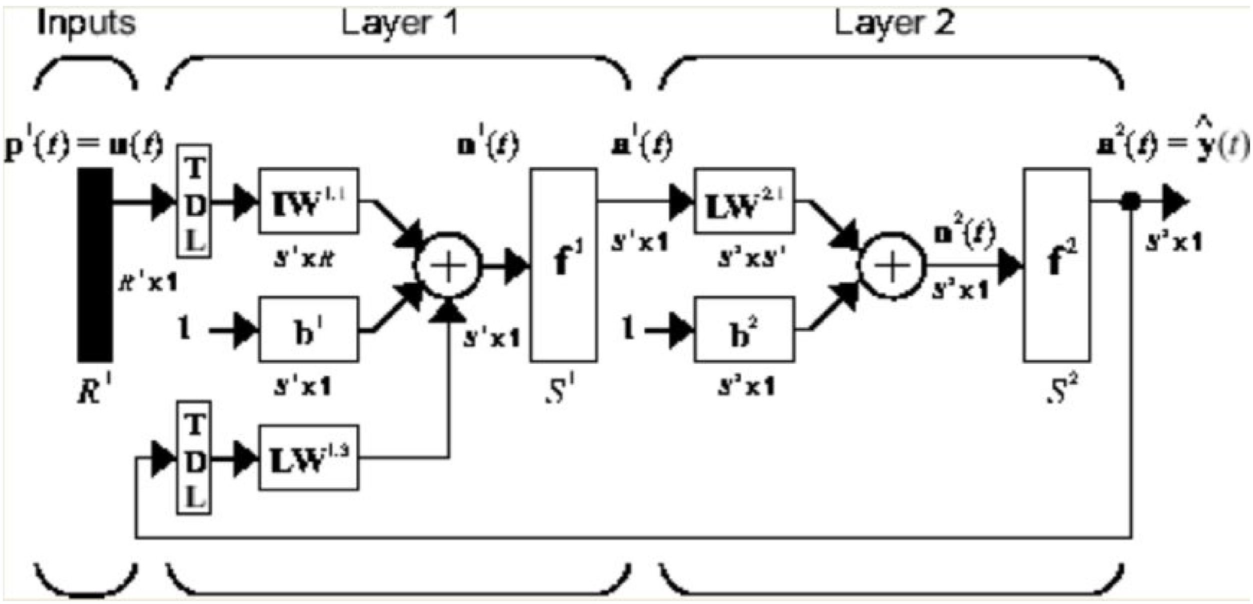

Figure. 4.2. NARX Block Diagram representation

4.2.2 Non-linear Input Output (NLIO):

The NLIO has a difference with respect to the NARX because in NLIO the inputs are only the past values of x and hence we determine the value of y in absence of a feedback or an error network in case of NLIO. The NLIO network comprises of tapped delay lines and a layer feedforward network with a sigmoid transfer function in the hidden layer and linear transfer function in the output layer. NLIO can be seen as a simple feedforward network, as shown in fig.4.3. This network time series is generally defined as follows :

$$y_t = F(x_{t-1} , x_{t-2} , x_{t-3} , x_{t-4} , \ldots\ldots , x_{t-d}) \tag{16}$$

here,

y - variable of interest / target values

x - externally determined parameter

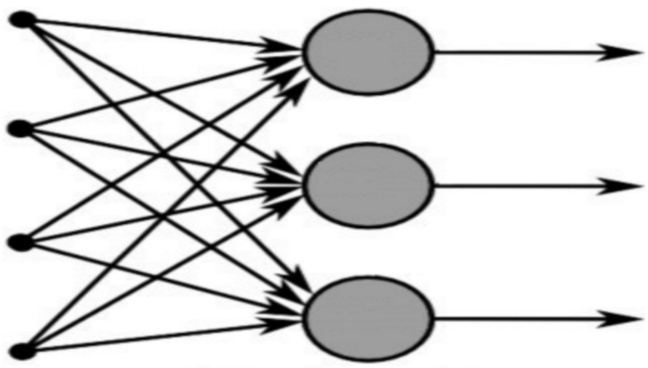

Figure 4.3. Feedforward Neural Network Representation

4.2.3 Recurrent Neural Network (RNN):

In the case of RNN, the processed data moves via loop or feedback. If a decision is made, it considers the current input and also what it has memorized from the inputs it received in the past. RNN adds immediate past to the present. Recurrent Neural Network has multiple sources of info, the present and the ongoing past. A simple or basic RNN can be depicted by fig. 4.4.

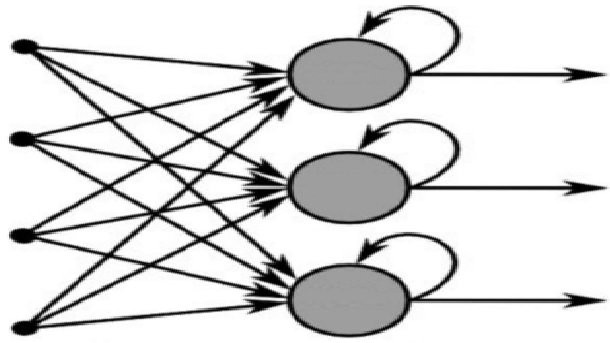

Figure 4.4. Recurrent Neural Networks (RNN) Basic Representation

In a much simpler terms , we can show with the help of following equations , how RNN can be evolved over a time :

$$o^t = f (h^t ; \theta) \tag{17}$$

$$h^t = g (h^{t-1}, x^t ; \theta) \tag{18}$$

here, the variables used are as follows:

o^t = output of the recurrent neural network at time t

x^t = Input to RNN at time t

h^t = state of hidden layer(s) at time t

θ = it encapsulates the weight and bias of the network

The relation between the above parameters can be shown by following fig. 4.5 :

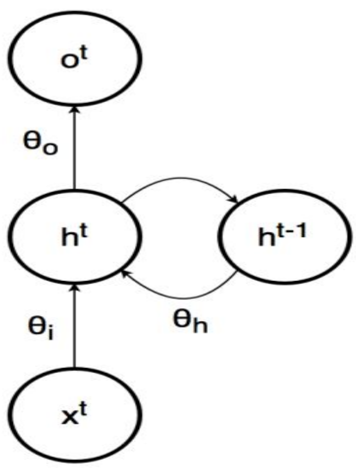

Figure 4.5. Graphical representation of RNN model

equation (17) says: : variable θ, output at the instant of time (t) depends upon the situation of hidden layers at time (t) , which is quite similar to that of feedforward neural network.

equation (18) says : variable θ, which is a hidden layer at the instant of time (t), it is dependent on ,the hidden layer at time (t-1) and the input at the instant t.

This equation (18) shows that the RNN can retain information of its precedent by allowing past values h^{t-1} to influence its present computation h^t.

Therefore, the aim of training RNN is to obtain sequence $o^{t+\tau}$ so that it can match sequence , $y_{t.}$

Here , τ is the time lag which exist among first significant RNN output $o^{\tau+1}$ and the first target output y_t.

This is comparable to the case that how we human being transform English to French , which frequently starts by understanding the initial few words so that it can be provided to the system with the context in order to translate the rest of the sentence .

4.3 Results and Discussions

The simulation was performed using MATLAB R16A software. Bayesian regularization (BR) [4][3] was used for training the neural network. It is an arithmetical process which transform nonlinear-regression to a distinct statistical numerical problem in the order of a crumple regression. Bayesian regularized artificial neural networks (BRANNs)[4][3][20] are further energetic than back-propagation network (BPNs) [20] and it can eradicate the requirement of a long-lasting cross-validation. It reduces the amalgamation of squared errors (e^2) and weights (w), then find out the accurate mixture so as to produce a network that generalizes well. The benefit of BR artificial neural network is its capability to disclose potentially complex relationships sense which can be used in quantitative studies to make a robust model. In general, Bayesian-Regularization (BR)[3][4] requires more time as compared to Levenberg-Marquardt (LM)[3] but it can result in good generalisation for difficult, noisy and small sets of data. The network complete its cycle when the target MSE was achieved or following the maximum number of epochs was reached.

After the training was completed the testing was done on different data set and errors were plotted to evaluate the performance of the designed network. The performance parameters which are utilised for comparing the neural networks are discussed below.

4.3.1 Performance Parameters :

4.3.1.1 Mean Square Error (MSE):

The MSE[9] is defined as the second moment (about the origin) of the error, and thus includes the variance of the predictor and the variance of its bias. Mathematically, MSE can be represented by eqn. 19.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 \tag{19}$$

$\widehat{Y}_i$= Predicted values; Y_i= Dependent variable ;

4.3.1.2 Mean Absolute Error (MAE):

According to Statistical Mathematics MAE is the determination of the difference between the two continuous variables. The Mean Absolute Error (MAE)[9] is given by eq.20 .

$$MAE = \frac{\sum_{i=1}^{n}(y_i - x_i)}{n} \tag{20}$$

Where y_i is the predicted value and x_i is the observed/ true value. The MAE is a common way to determine forecast error in time series analysis.

4.3.1.3 Mean Absolute Percentage Error (MAPE):

It is an assessment of accuracy of a forecasting method. It basically, implies accuracy as a percentage. MAPE[9] is defined by eq. 21.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{21}$$

Where A_t is the actual value and F_t is the forecast value.

4.3.1.4 Root Mean Square Error (RMSE):

It is the evaluation technique by which we measure the difference between forecasted values by a model and the observed values. RMSE[9] is the measure of ACCURACY, commonly used to compare forecasting errors of different predictive models , for the same data-set. Mathematically, RMSE is expressed as eq. 22 .

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}} \tag{22}$$

Where ,

$\hat{y}_t$ = Predicted values ; y_t = dependent variable ;

T = no. of data in dataset

In this section the results of the simulation are presented and discussed through tabular comparative study for all the three models. Fig. 4.6, fig. 4.7 and fig. 4.8 shows the MSE error plot during training for NARX, NLIO and RNN respectively. It shows that the training of RNN has completed only in 6 epochs with minimum error as compared to other two networks. Fig. 4.9, fig. 4.10 and fig. 4.11 shows RMSE error plot in short-term wind power forecasting during testing for one day using NARX, NLIO and RNN respectively. It shows that the RMSE error is minimum for RNN as compared to the other methods. However peak in the error is occurring at the same time.

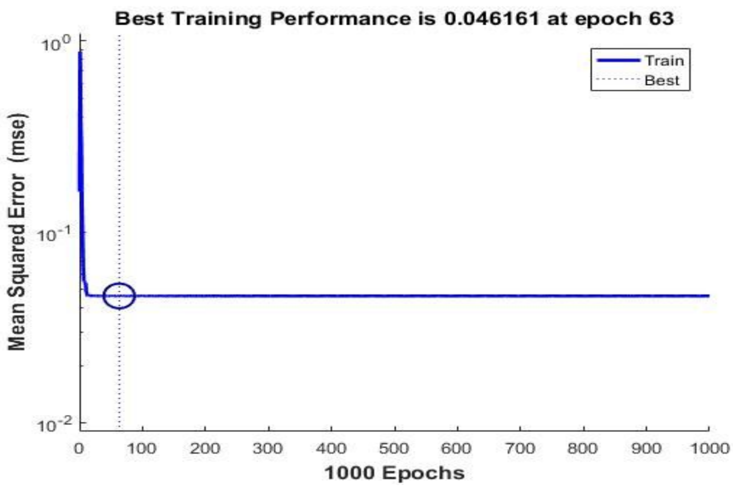

Figure. 4.6. Plot of MSE for NARX network

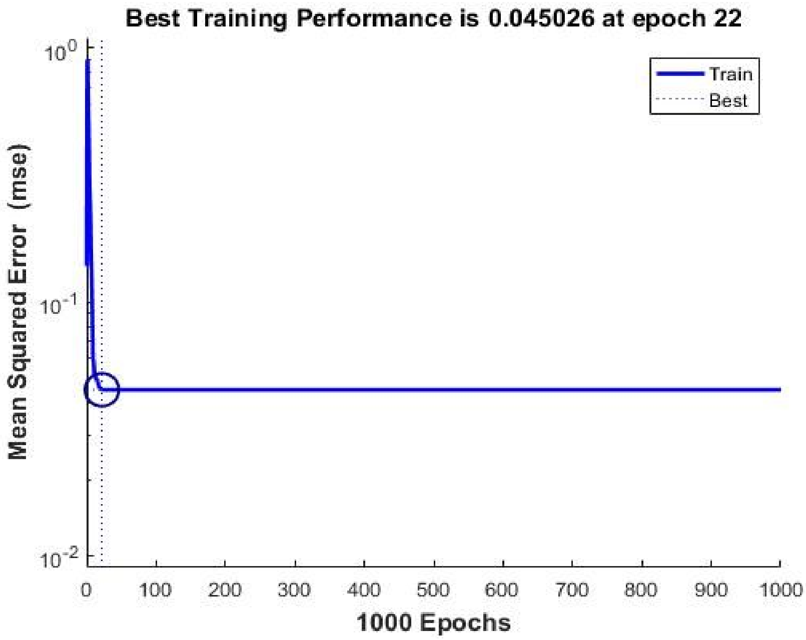

Figure. 4.7. Plot of MSE for NLIO network

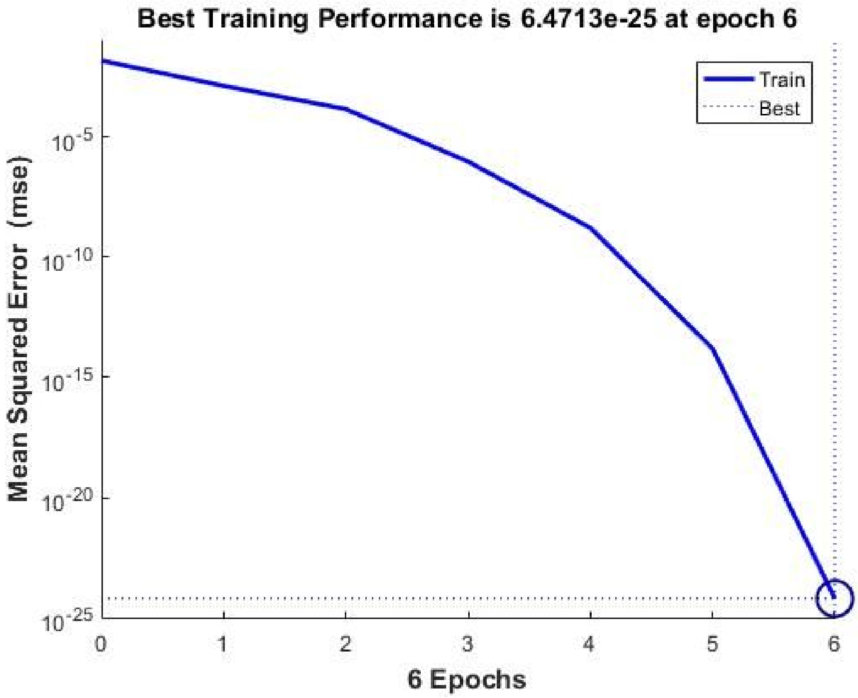

Figure.4.8. Plot of MSE for RNN network

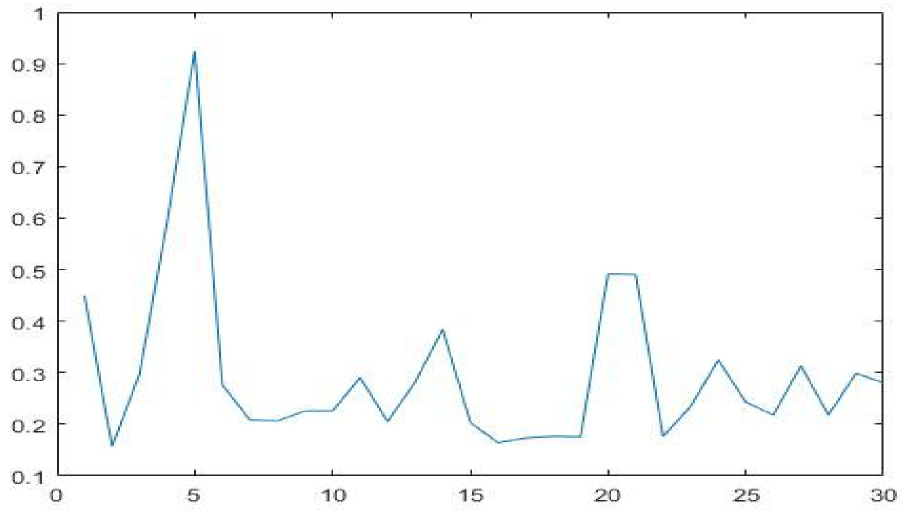

Figure. 4.9. Plot of RMSE for NARX network

The above plot for NARX is between RMSE value on y-axis and Time period on x-axis.

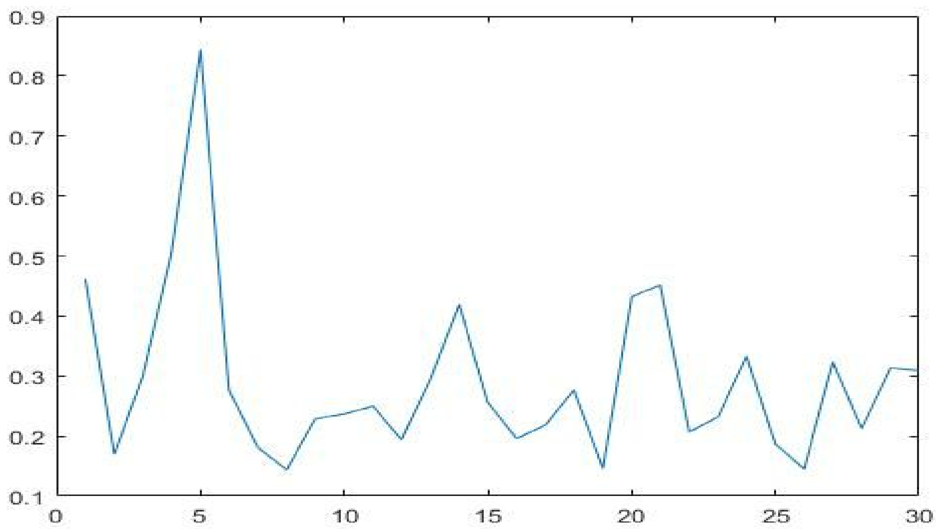

Figure. 4.10 RMSE Plot for NLIO network

The above plot for NLIO is between RMSE value on y-axis and Time period on x-axis

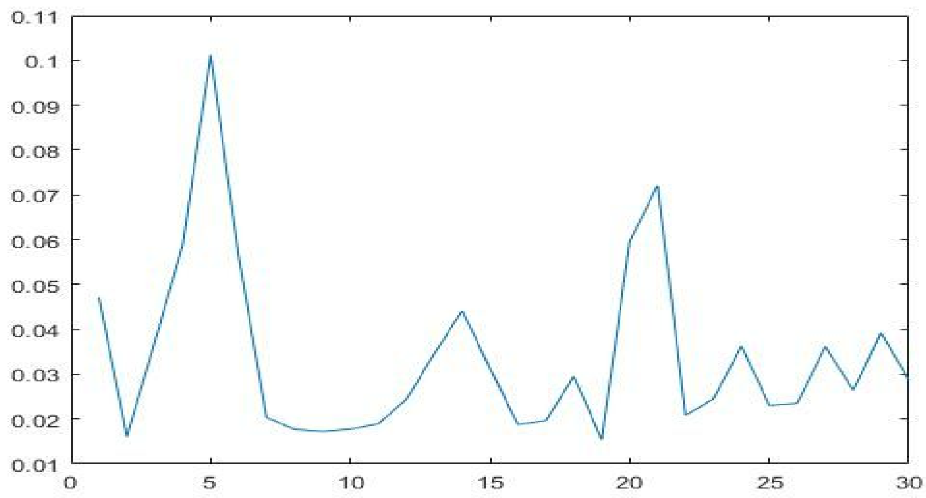

Figure. 4.11 Plot of RMSE for RNN network

The above plot for RNN is between RMSE value on y-axis and Time period on x-axis

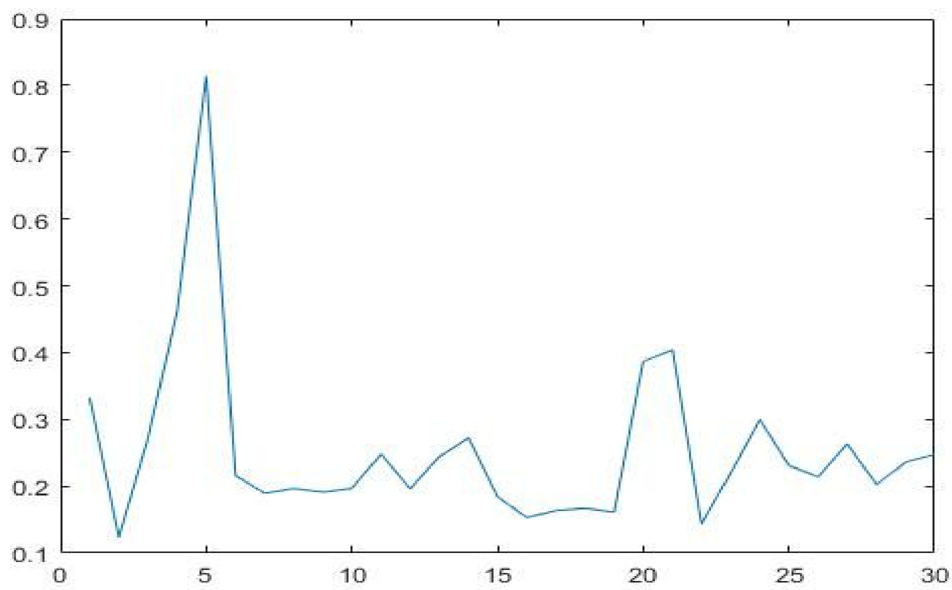

Figure. 4.12. MAE Plot for NARX network

The above plot for NARX is between MAE value and Time period.

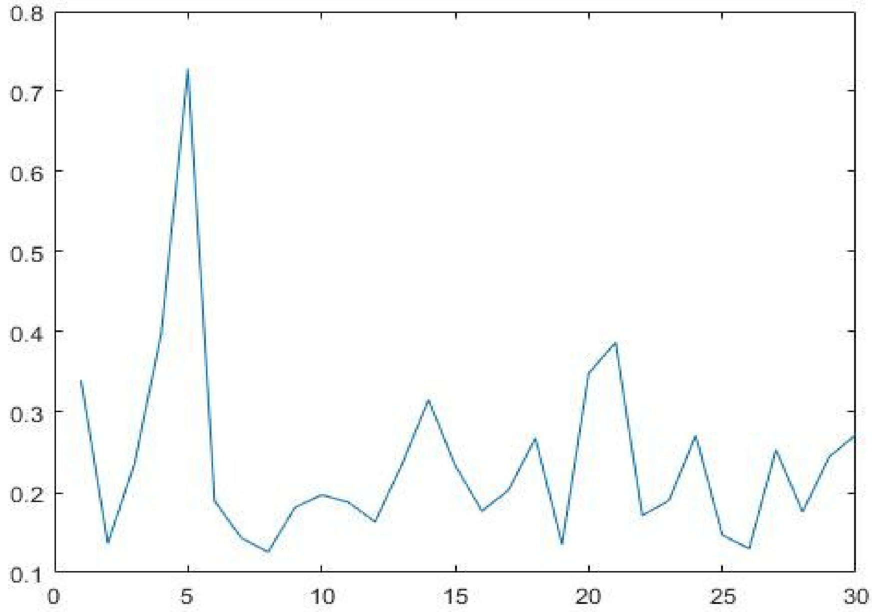

Figure. 4.13 MAE Plot for NLIO network

The above plot for NLIO is between MAE value and Time period.

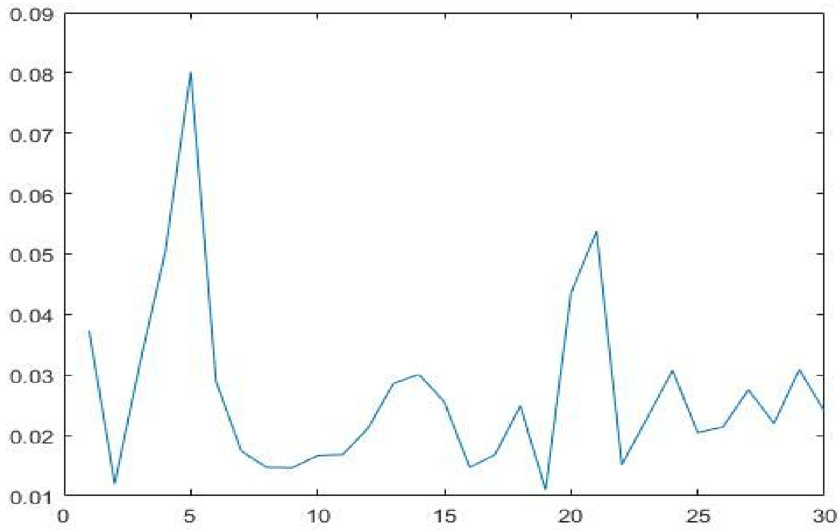

Figure. 4.14 MAE Plot for RNN network

The above plot for RNN is between MAE value and Time period.

The MAE plot shown by fig. 4.12, fig. 4.13 and fig. 4.14 whereas MAPE plots are shown by fig. 4.15, fig. 4.16 and fig. 4.17 for all the three networks. The MAE and MAPE are also minimum for RNN as compared to other two networks.

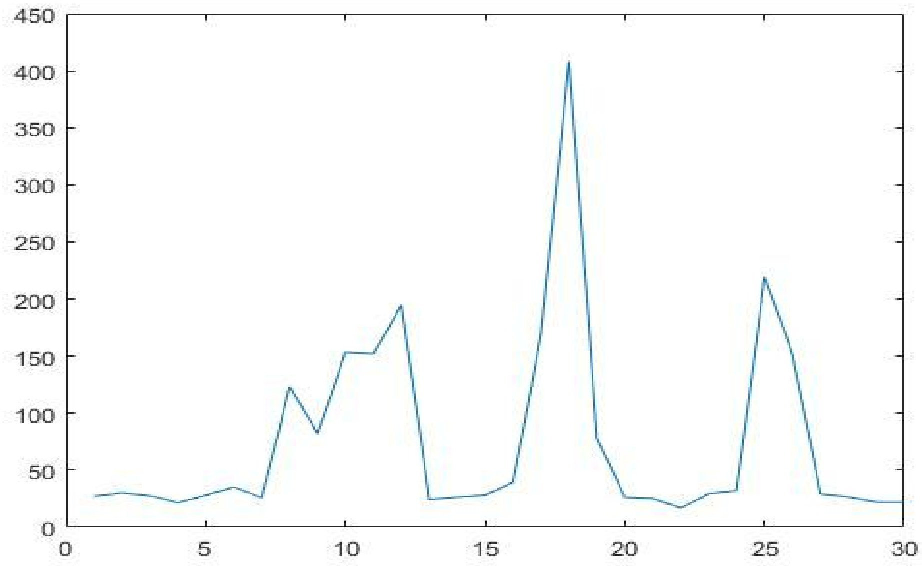

Figure. 4.15 MAPE Plot :NARX network

The above plot for NARX is shown between MAPE value on y-axis and Time period on x-axis

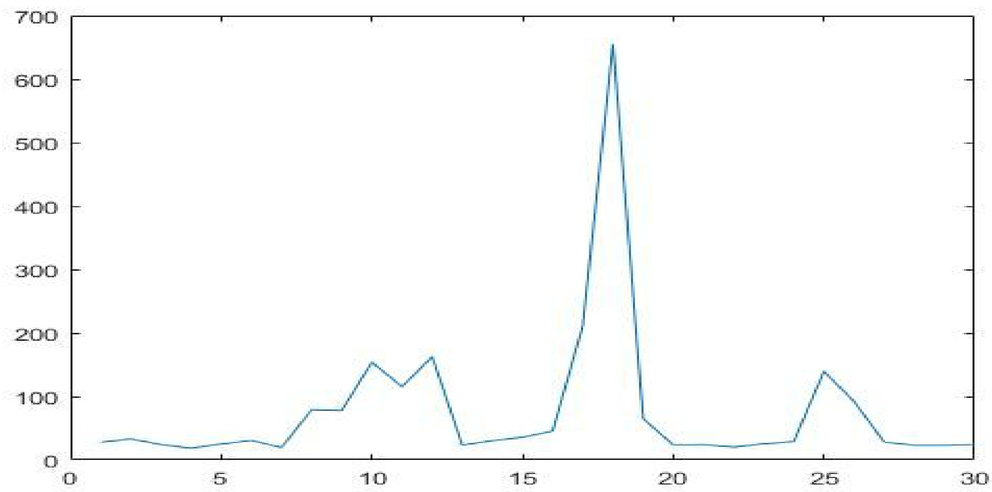

Figure. 4.16 MAPE Plot : NLIO Network

The plot for NLIO is made between MAPE value on y-axis and Time period on x-axis.

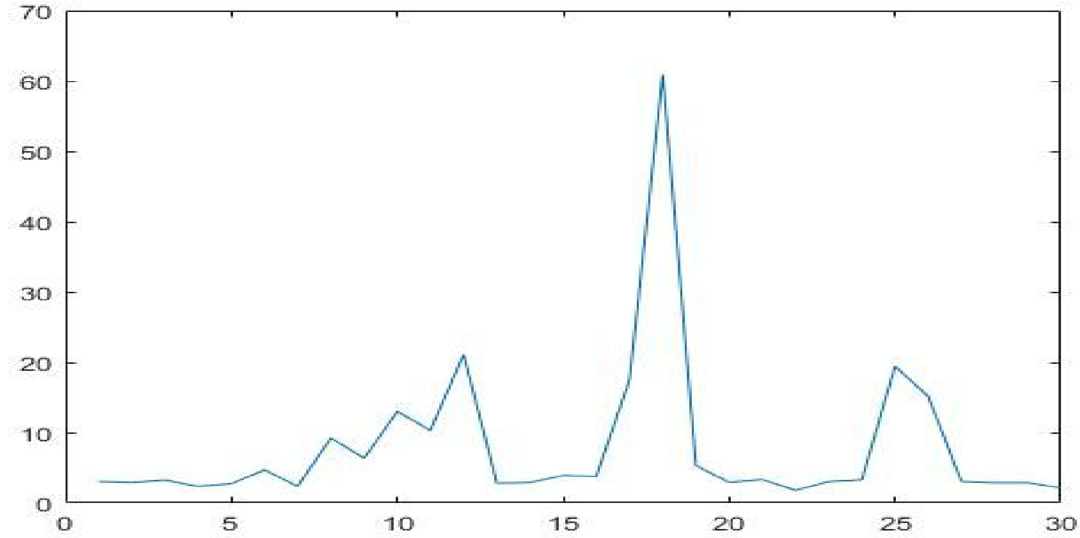

Figure. 4.17 MAPE Plot : RNN network

The above plot shown for RNN is between MAPE value on x-axis and Time period on y-axis.

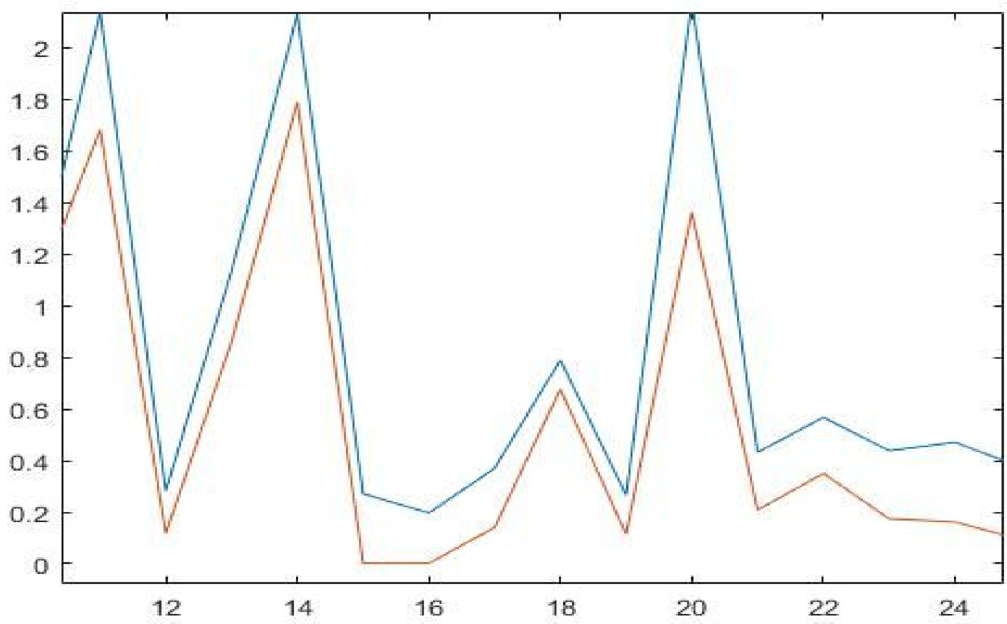

Figure. 4.18 Predicted vs actual load using NARX

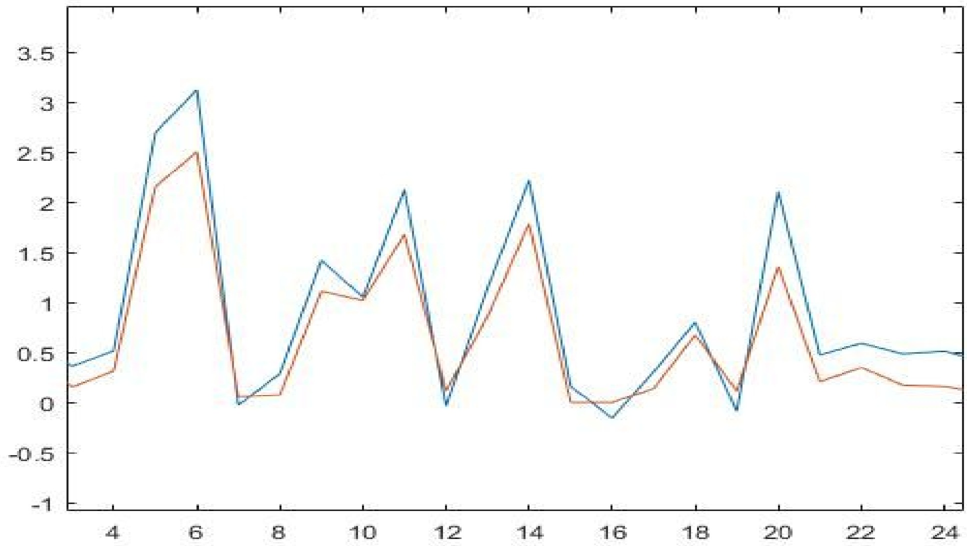

Figure. 4.19 Predicted vs actual load using NLIO

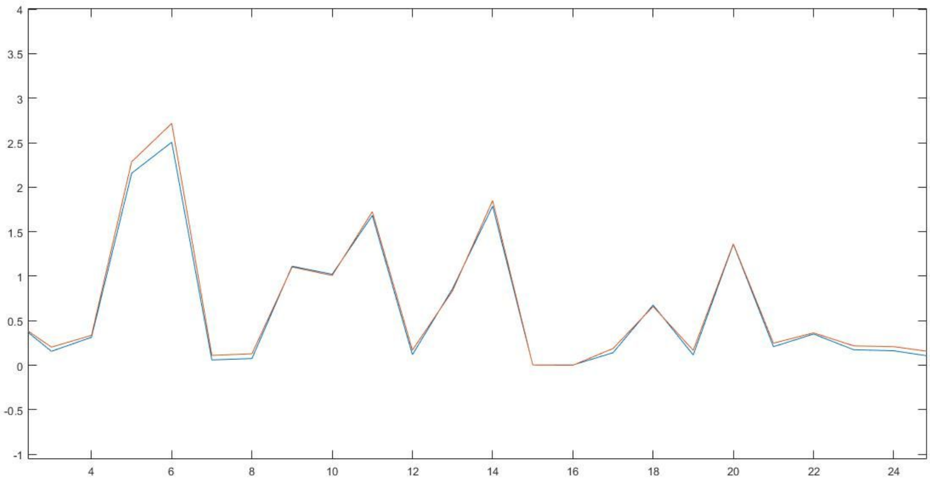

Figure. 4.20 Predicted vs actual load using RNN

Fig. 4.18, fig. 4.19 and fig. 4.20 shows the actual and forecasted wind power generation for 24 hours. Here on x-axis is the time period while on y-axis we have taken the actual and predicted values of our targeted parameter Very obviously the RNN shown that the forecasted wind power follows the actual one closely. All the results are presented in table 2 in a comparative manner which shows that

Table 2. Table showing the comparative analysis of all the neural networks

PARAMETERS	NARX_NETWORK	NLIO_NETWORK	RNN_NETWORK
Mean Absolute Error (MAE)	0.25419	0.23944	0.026856667
Mean Absolute Percentage Error (MAPE)	75.843426	76.21588	7.94474667
Mean Square Error (MSE)	0.46161	0.045026	6.4713e-25
No. of epoch	63	22	6
Root Mean Square Error (RMSE)	0.29696	0.291376	0.03388667

From above table following conclusion can be made :

1) RNN outperforms NARX and RNN, it can be observed from MAPE value that RNN have much lower value for RNN than NARX and NLIO.

2) The number of iteration or epoch taken by the RNN is quite lesser than that of NARX and NLIO.

3) NARX model as well as NLIO model cannot be used for time series forecasting as they are having a very high value of error as compared to RNN algorithm.

CHAPTER 5

PREDICTIVE ANALYSIS USING GBM, LSTM AND RNN

5.1 Introduction

In this chapter we study and implement three techniques : a Basic Recurrent neural network (RNN) , Gradient Boosting Machine (GBM)[23] contrasted with LSTM technique , all are used for the prediction of wind turbine power output, using the wind velocity as a input parameter. Then , all the three models are compared with each other using certain performance parameters and analysis is done based on the outcome of these parameters. Furthermore , the best model is suggested based on the outcomes. Also, the scope of future work is there regarding improvement in the accuracy of Neural system by using various other hybrid techniques along with both the models. In this case, dataset used , contains annual hourly data of wind turbine based in Kolkata region of India, having wind velocity and turbine Power output as the two parameters.

The mathematical relationship between wind speed and power output is given by :

$$P_{wind} = (1/2)\,\rho\, A v^3 C_p \tag{23}$$

where,

P_{wind} is the output power of a wind turbine (kW)

A is the wind turbine Rotor area (m^2)

ρ is the air density (kg/m^3)

v is the velocity of wind (m/sec)

C_p is the utilization factor of wind energy

As from eq. 23, it is clear that the power generated from wind turbine is mainly dependent on velocity of wind , since all other parameters such as rotor of wind turbine, air density etc are constant for a given specific conditions. So in order to forecast the wind turbine output power, the wind data which is recorded on hourly basis is considered.

The Block diagram for the whole process can be shown as in Fig. 5.1.

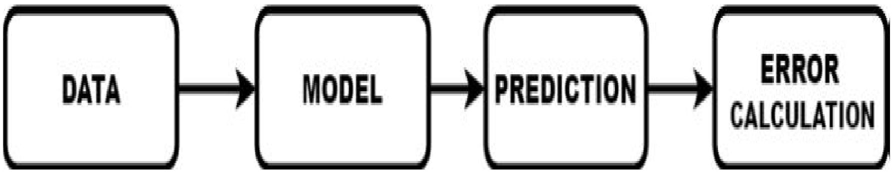

Figure 5.1. Block Diagram of Process

5.1.1 Preprocessing of Dataset

The data of Kolkata region in India has been used for wind power forecasting. The data considered in this paper is the hourly data of 2014 (1st Jan to 31st December). This data set was split into the 24-hours data each subset, thus grouping them for 365 days. This dataset was then divided into training set and testing set. The train size and test size is 70% , 30% respectively of the annual hourly based dataset , which is used for the forecasting of power output of wind turbine with the help of wind velocity.

5.2 Algorithm Description

RNN, GBM and LSTM algorithms are used in this work as described below, with following features:

a) The size of hidden layer for each network is 100 .

b) In all the algorithms, Wind velocity (in m/sec) is Input parameter, while the targeted parameter or the output parameter to be predicted is Output Power (in kW).

5.2.1 RNN - Recurrent Neural Networks (RNN) are quite old, in the similar retro fashion of that other famous learning calculation. In the year 1980's RNN were first introduced, and since then, they are showing their real potential from past decades, in the field of machine learning and artificial intelligence RNN can be shown in general by Fig. 5.2.

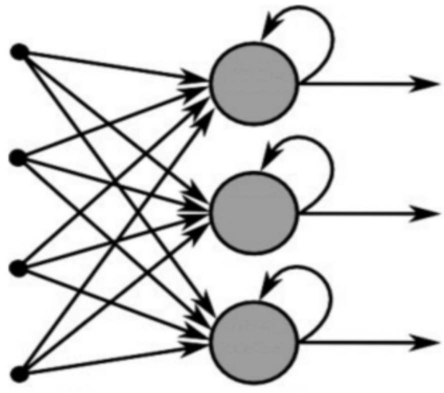

Figure. 5.2. General Representation of RNN

RNN is a category of neural networks where links flanked by the computational units led to formation of a directed circle. Contrasting to the MLP or feedforward network, BRNN know how to utilize their in-house remembrance or memory to procedure out the random sequences of inputs. Each one of the constituting unit of a RNN is consisting of a time changing real valued activation function and also changeable bias and weight. RNNs therefore are shaped by the application of the identical sets of weights which are recursively over a graph-like structure.

In a RNN, the information or data, cycles in the course of a loop. When it construct a decision, it takes into account the current input as well as what it has well-read from the inputs it received previously. As a constructive and inherited feature of their inside memory, Recurrent Neural Network (RNN's) can recall vital actions regarding the retrieved information which they get, and hence empowers themselves to be exceptionally accurate in anticipating that next thing is what's coming straightaway. *A Recurrent Neural Networks(RNN) adds immediate past to the present.* RNN's has multiple sources of info, the present and the ongoing past. In a much simpler terms , we can show with the help of eq. 24 and eq.25 , how RNN can be evolved over a time :

$$o^t = f\,(h^t \,; \theta) \tag{24}$$

$$h^t = g\,(h^{t-1}, x^t \,; \theta) \tag{25}$$

where, the variables used are as follows:

o^t = Output of the Recurrent Neural Networks at time t

x^t = Input to RNN at time t

h^t = state of hidden layer(s) at time t

θ = it encapsulates the weight and bias of the network

The relation between the above parameters can be shown by Figure 5.3.

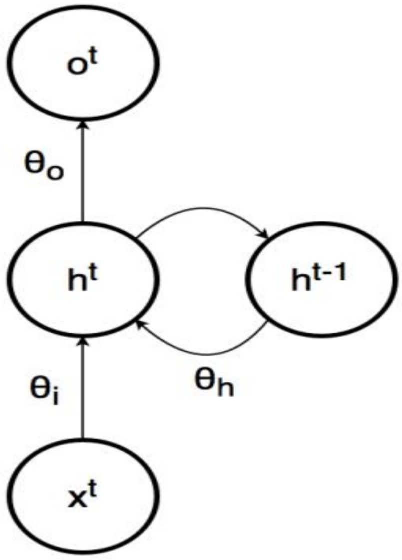

Figure 5.3. Graphical representation of RNN model

Many of the RNNs use to define the values of their hidden units , as in eq. 25.

equation (24) says : variable θ, output at the instant of time (t) depends upon the situation of hidden layers at time (t) , which is quite similar to that of feedforward neural network.

equation (25) says : variable θ ,the hidden layer at the instant of time (t) depends upon ,the hidden layer at time (t-1) and the input at the instant t.

The eqn. 25 represents that RNN can retain information of its precedent by allowing past values h^{t-1} to influence its present computation h^t.

Therefore, the aim of training RNN is to obtain sequence $o^{t+\tau}$ so that it can match sequence , y_t..

Here , τ is the time lag which exist among the first significant RNN output $o^{\tau+1}$ and the first target output y_t.

In the case of RNN, the model prepared after thorough learning has the identical size of input, since it is precise in terms of evolution from one state to another state. At the same time the algorithm utilises the same transition or evolution function with the identical parameters at every time step.

5.2.2 **LSTM-** LSTM is an unusual sort of RNN, evolved in 1997 by Hochreiter and Schmidhuber. On observing its architecture, we deduced that the standard hidden layers of BRNN are substituted by LSTM cells. These cells are made up of a variety of gates which can manage the input flow. A LSTM cell comprises of an output gate, forget gate, cell state, and a input gate. LSTM cells also comprises of tanh layer, sigmoid layer and point wise multiplication function. The diverse gates and their functional features are given below:

- Input gate : It comprises of input information.
- Cell State : It covers the whole network and is capable of adding or removing information or data with the support of other gates.
- Forget gate : It defines and recognise the fraction of the information to be allowed.
- Output gate : It comprises of the outcome function or basically the output, generated by the LSTM.
- The function of Sigmoid layer is to generate numerals ranging between zero and one, deciding what percentage of each component should be let through.
- The Tanh layer creates a new vector, which finally will be added to the state.

The cell state is accordingly reorganized depending upon the outputs forming the gates. Mathematically, LSTM cells can be represented by following equations eq.(26), (27),(28),(29),(30).

$$f_t = \sigma\,(W_f\,[h_{t-1}\,,x_t] + b_f\,) \tag{26}$$

$$i_t = \sigma\,(W_i\,[h_{t-1}\,,x_t] + b_i\,) \tag{27}$$

$$c_t = tanh\,(W_c\,[h_{t-1}\,,x_t] + b_c\,) \tag{28}$$

$$o_t = \sigma\,(W_o[h_{t-1}\,,\,x_t] + b_o\,) \tag{29}$$

$$h_t = o_t * tanh\,(c_t) \tag{30}$$

where,

h_t : Output Vector ; x_t : Input Vector ; c_t : Cell State Vector

i_t : Input gate Vector ; f_t : forget gate vector

o_t :Output gate Vector ; and W , b are the parameter matrix and vector. The basic LSTM using the above eq.(26), (27), (28), (29), (30) can be shown in below Fig. 5.4.

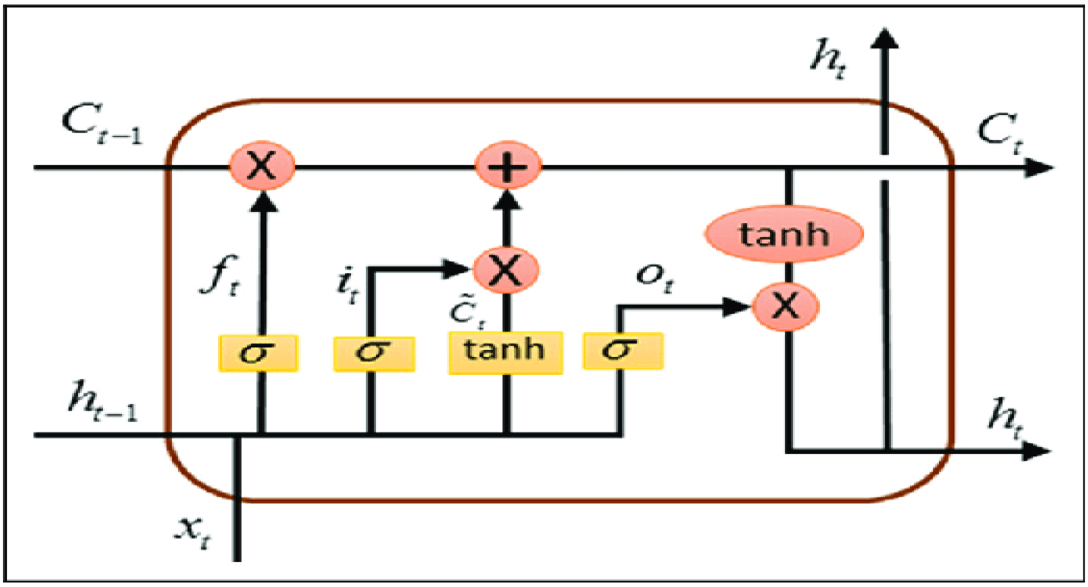

Figure 5.4. Internal Diagram of LSTM

5.2.3 **<u>GBM</u>** **:** GBM(Gradient Boosting Machine) is a methodology for resolving classification and regression problems, which generates a predictive or forecasting model in the variety of ensemble of weak predictive model, typical example is a decision trees. GBM constructs the model or system in a step- by-step or stage-wise manner, which is very much similar to other alternative boosting methodology do, and thus it generalizes them by permitting the minimization or optimization of an arbitrary differentiable loss function. Boosting is one type of ensemble technique where the forecasters/ predictors are not created alone, rather they are created sequentially.

Like other boosting methodology, gradient boosting joins the less strong "learners" into a single strongest learner in a unique iterative way. It is easiest to clarify in the least-squares regression setting, where the endeavor is to "teach" a function or model F which have to predict values of the form $\hat{y} = F(x)$ by reducing the mean square error (MSE) shown in eq. 34.

If M is the maximum value of the boosting gradient, at every stage m of (1<m<M) , of gradient boosting, then it might be certain that there is some defective model, Fm, so the gradient boosting calculation enhances Fm by developing another strategy/model that adds an estimator h, to give a superior model as given by eq. 31.

$$F_{m+1}(x) = F_m(x) + h(x) \tag{31}$$

For calculating *h*, the gradient boosting iteration leap forward having the observation that guarantees perfect *h* would be obtained by eq. 32.

$$h(x) = y - F_m(x) \tag{32}$$

Where $y = F_{m+1}(x) = F_m + h(x)$

Therefore, gradient boosting will try to fit the function *h* to the residual as given by eq. 33

$$\text{Residual} = y - F_m(x) \tag{33}$$

In the boost gradient machine, each F_{m+1} tries to correct the error of its predecessor F_m.

5.3 Results and Discussions

In this section the result of the simulation are presented and discussed through tabular comparative study for all the models , that is RNN, LSTM and GBM. All the models are run for same number of epochs, and also having same number of input vector size , hidden layers , output vector size. Now below Fig. 5.5 shows the forecast vs actual plot for Basic RNN (BRNN) and Fig 5.6 shows the MAPE plot for RNN Also Fig. 5.7 shows the forecast vs actual plot for LSTM and Fig 5.8 shows the MAPE plot for same algorithm. While Fig. 5.9 shows the forecast vs actual plot for GBM and Fig 5.10 shows the MAPE plot for GBM.

The graphs or plot are made between y_true and y_pred values. Where, y_true are the True or actual value of the target/output while y_pred is the forecasted or predicted value of the target/output.

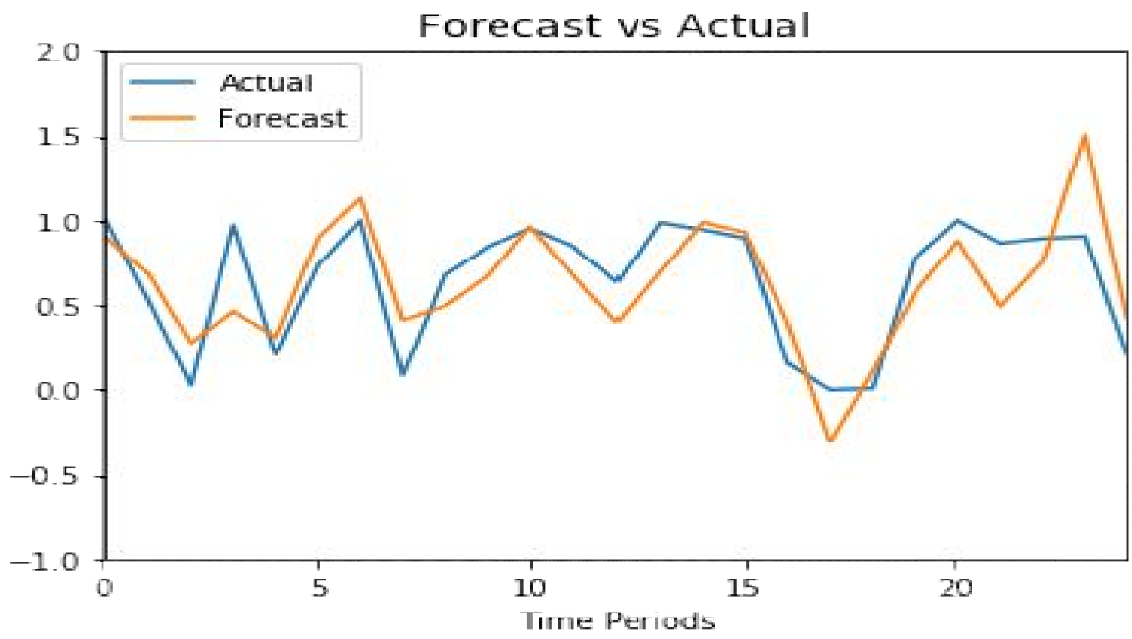

Figure. 5.5 Forecast vs Actual graph for RNN

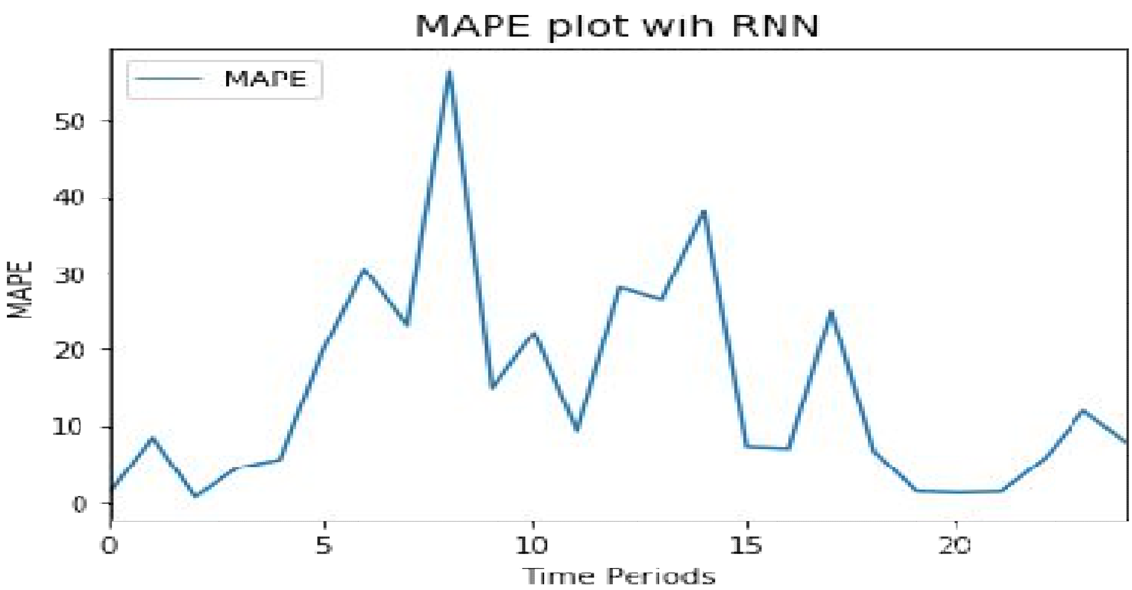

Figure. 5.6 MAPE plot with RNN

Now , for the LSTM all the figures of results , that is Forecast vs Actual time series , bar plot of forecasted and actual , along with the MAPE plot are shown as follows:

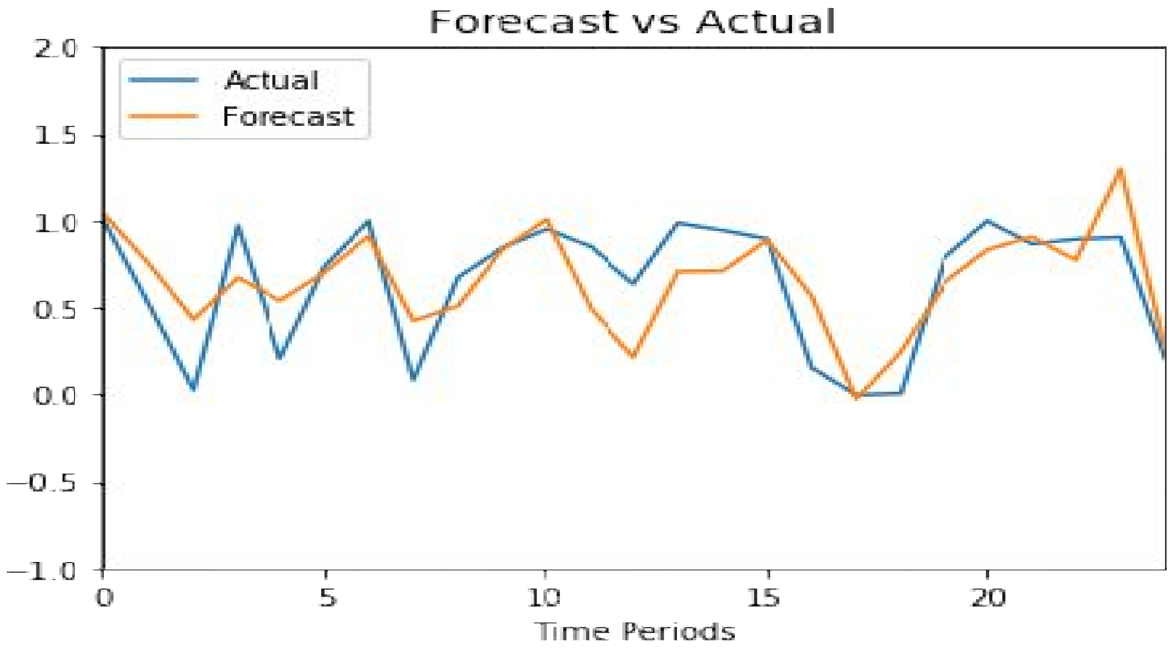

Figure.5.7 Forecast vs Actual graph for LSTM

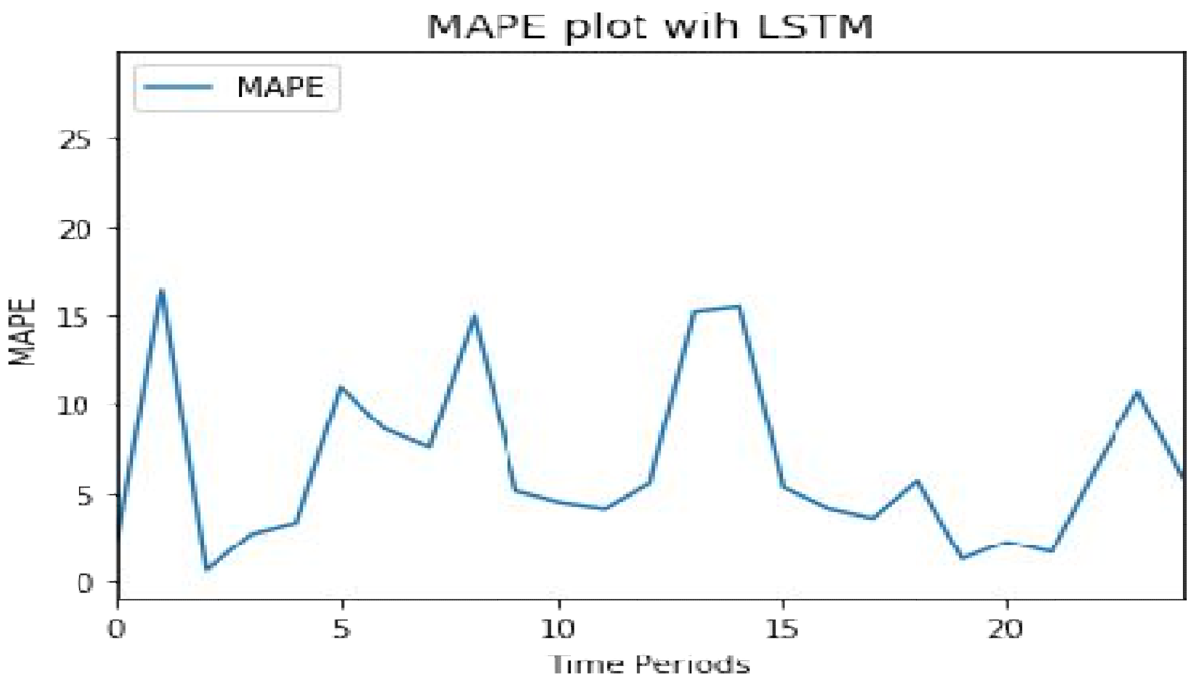

Figure. 5.8 MAPE plot with LSTM

For, GBM , all the figure of results , that is Forecasted vs Actual Time series, bar plot of forecasted vs actual, along with the MAPE plot are shown as follows :

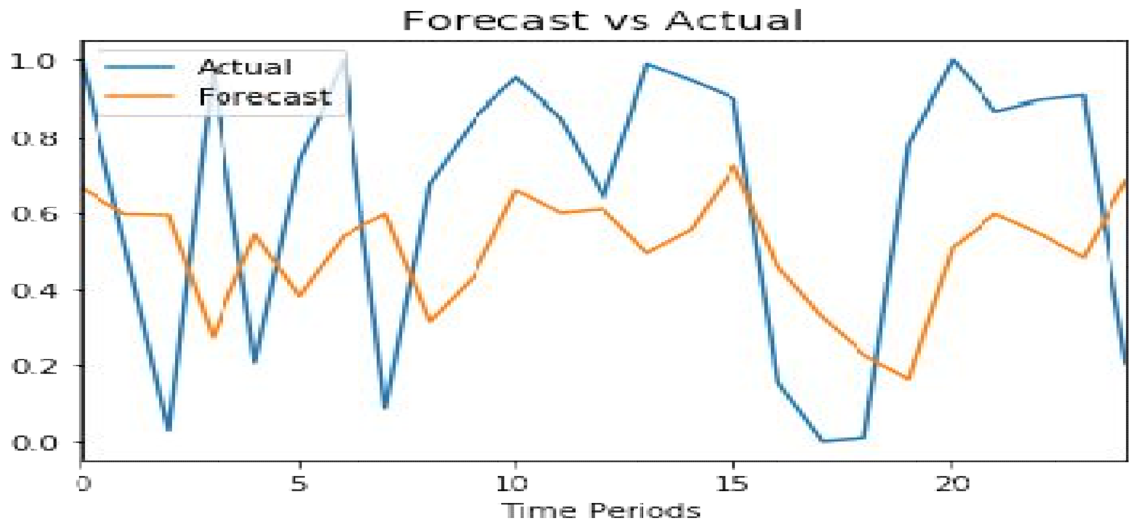

Figure. 5.9 Forecast vs Actual graph for GBM

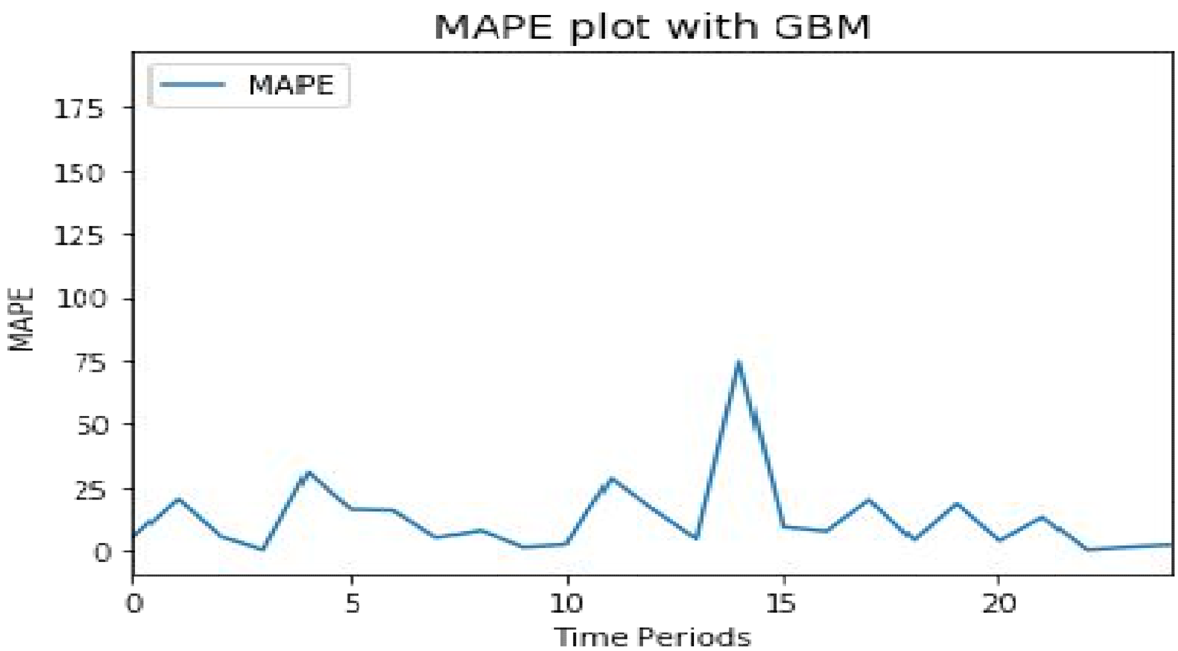

Figure. 5.10 MAPE Plot with GBR/GBM

Now on basis of performance parameters, all the algorithms are compared numerically, as shown in the below Table 3.

Table 3. Comparative analysis of Basic RNN, LSTM and GBM

PARAMETERS	RNN	LSTM	GBM
Mean Absolute Error (MAE)	0.264405	0.176683	0.320215
Mean Absolute Percentage Error (MAPE)	9.867345	6.563425	11.1049
Mean Square Error (MSE)	0.0080	0.0078	0.1623
Root Mean Square Error (RMSE)	0.354985	0.231292	0.402920

From above comparative table the observations are as follows :

(i) It is observed that the LSTM have lower values of MAE,

as compared to that of RNN and GBM.

(ii) MAPE is least for LSTM as compared to that of RNN and the highest MAPE value is for GBM.

(iii) MSE and RMSE values are least for LSTM and highest for GBM, while its intermediate value for RNN.

Thus, it clearly proves that the final outcomes and performance of LSTM is better than RNN and GBM.

CHAPTER 6

PREDICTIVE ANALYSIS OF WIND TURBINE POWER OUTPUT USING GA BASED SVM

6.1 Introduction

This chapter displays a novel methodology of GA based SVM[24-25-26] hybrid[36] algorithm, where Genetic algorithm is used for feature selection, and SVM is the main predictive model. Basically, we are using three input parameters viz. day, time and wind velocity to predict the output parameter that is power output[31-35] of wind turbine. Then, this hybrid[35] model i.e. GA based SVM is further benchmarked by comparing with other neural network models[30] like : RNN[28-29], Linear Regression, MLP/ANN. Further, with the help of performance parameters, all the algorithms are compared and the best method is suggested based on the outcomes of comparative results. Also, the scope of future work is there regarding improvement in the accuracy of hybrid model by using various other functions. In this case, dataset used, contains annual hourly data of wind turbine based in Kolkata region of India, having wind velocity and turbine Power output as the two parameters. Here, daily average[27] methodology is used to segregate the data for the work process.

The block diagram for the whole process can be shown in below Fig 6.1.

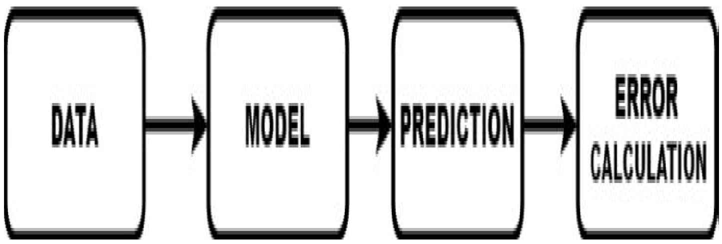

Figure.6.1. Block Diagram of Process

6.1.1 PREPROCESSING OF DATASET

The data of Kolkata region in India has been used for wind power forecasting. dataset, which is considered in this paper is the hourly data of 2014 (1st Jan to 31st December). This data set was then normalised between 0 and 1, and further the daily average of the data (according to each day) was found out. This dataset was then divided into training set and testing set. The train size and test size is 67% , 33% respectively of the normalised, daily average dataset, which is used for the forecasting of power output of wind turbine with the help of day, time, wind velocity. Fig. 6.2 shows pattern of annual dataset for wind power and wind speed. While Fig. 6.3 shows pattern for wind speed and Fig. 6.4 shows pattern for wind turbine power output or wind power.

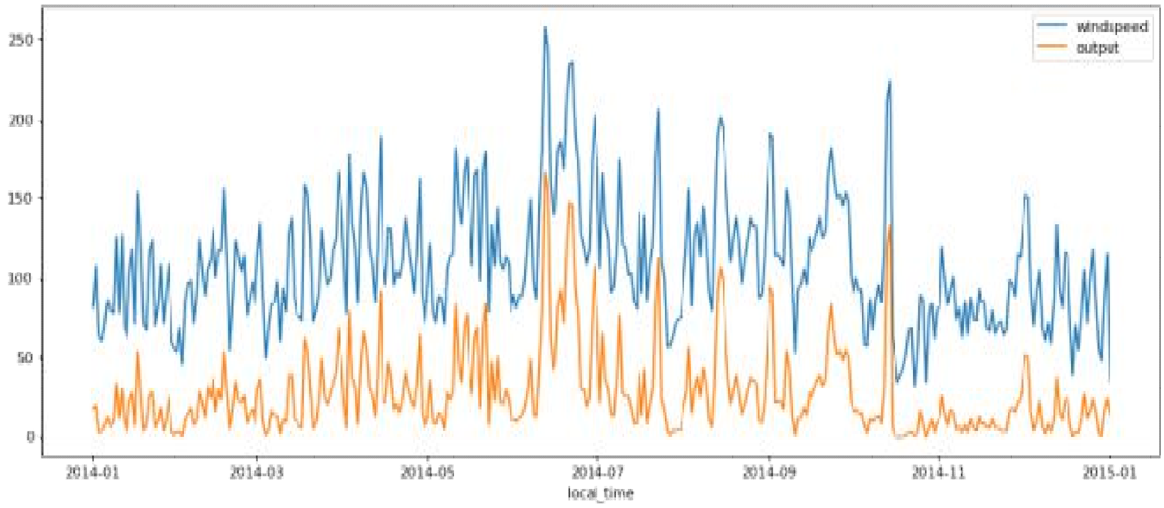

Figure. 6.2. Pattern of Annual Dataset

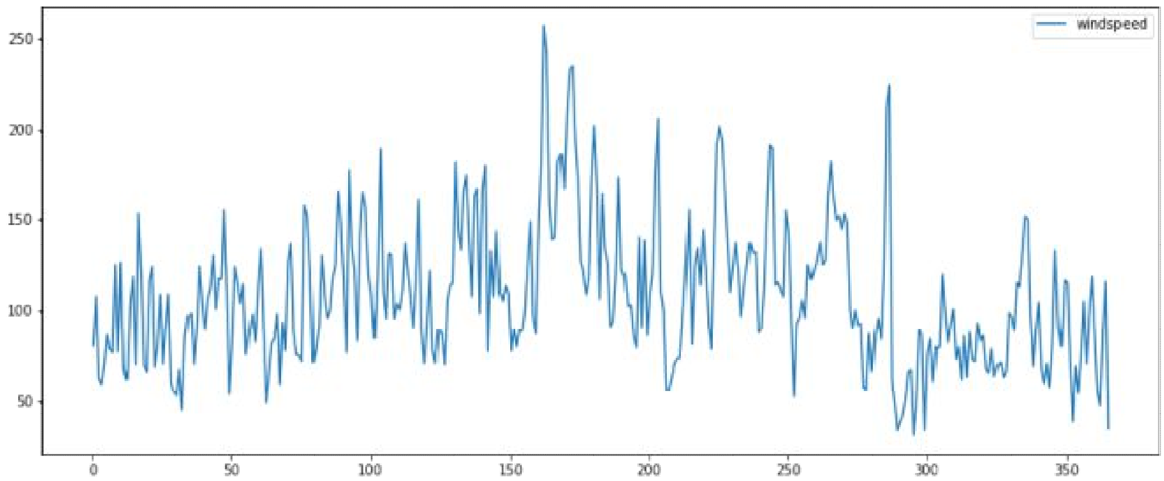

Figure. 6.3. Wind speed data pattern

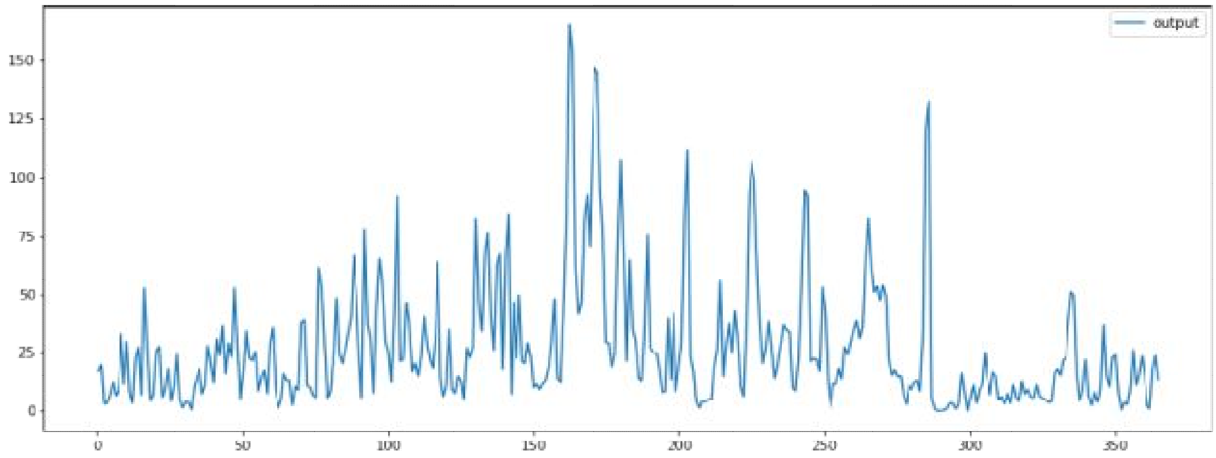

Figure. 6.4. Wind turbine output power data pattern

6.2 Algorithm Description

6.2.1 Support Vector Machine(SVM)

It is an algorithm fallen in the category of supervised machine learning, and its main feature is that, it can be used for both condition whether of classification or regression. In predictive analysis, we basically apply SVM for regression analysis. SVM as a classifier involves the process of plotting up of certain points from dataset in n-dimensional space (n is the number of features). Here, each and every characteristics represents return of single points or coordinate. Now thereafter the categorization is done by deducing out the hyper-plane which discriminate between the two classes accurately. Basic SVM can be pictorially shown in Fig 6.5

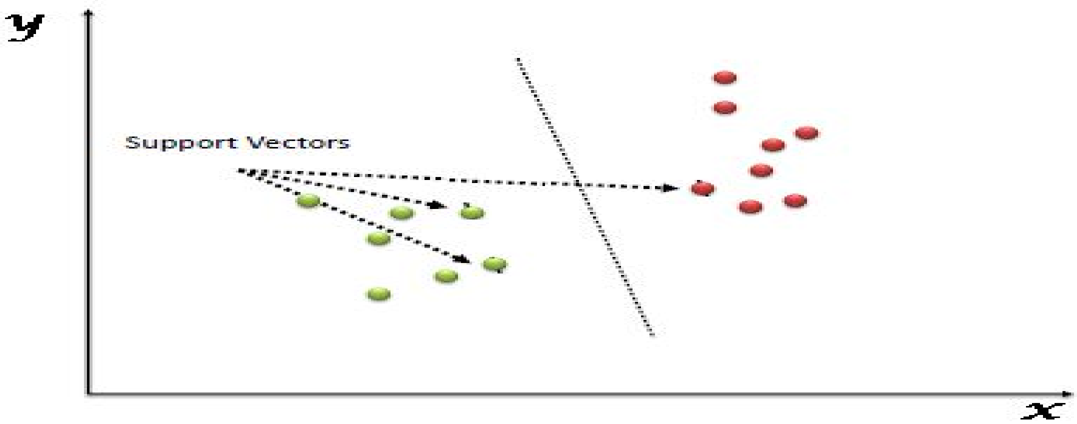

Figure. 6.5. SVM Basic representation

SVM as a regression analysis tool is of great use in case of Forecasting like : price forecasting , energy forecasting, power forecasting. The ability of SVM in resolving out the non-linear regression estimation problems, made it quite appropriate tool for forecasting. Mathematically, for describing the SVM, let us suppose training dataset as follows : $\{(\mathrm{X}_i, y_i)\}_{i=1}^{N}$ where , $X_i \in R^m$ is a vector input while y_i is its output. SVM approximating function can be shown as follows :

$$f(x,\omega) = \sum_{j=1}^{m} \omega_j \phi_j(x) + b \tag{34}$$

and the SVR will be solving a constrained optimization problem of minimizing a given function.

$$\min_{\omega,b,\xi,\xi^*} R(\omega,b,\xi,\xi^*) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{35}$$

subject to

$$y_i - (\omega^T\phi(X_i) + b) \le \varepsilon + \xi_i^*, (\omega^T\phi(X_i) + b) - y_i) \le \varepsilon + \xi_i, \xi_i, \xi_i^* \ge 0,$$

$i = 1,2,......N$

here, X_i is mapped to a higher dimensional feature space by the function ϕ. By doing such mapping, we basically are converting the non-linear regression problem in x-space into a linear regression problem related to ϕ-space. ξ_i and ξ_i^* are the slack parameters or variables, where, ξ_i^* is the upper training error and ξ_i is the lower training error, both subjected to the Vapnik's ε-insensitive loss function or tube

$\left|y - (\omega^T\phi(X) + b)\right| \le \varepsilon$. ω is the weight function and b is the bias. Now, here the quality of regression is guided by the researcher by selecting the legitimate values of following three parameters viz. cost error C, width of the tube ε and the mapping function ϕ.

Now, here one thing which we have to observe in equation 35 are the constraints, which implies that almost all data belonging to X_i is tried to maximum put inside the tube, $\left|y - (\omega^T\phi(X) + b)\right| \le \varepsilon$. So, here now following condition arises, If X_i lies within the tube, the loss is zero, otherwise if X_i is not inside tube then, there is an error which is either ξ_i or ξ_i^*, which is then minimized in the cost function C. The SVR (support vector regression) bypasses the over fitting and under fitting of the data, which is done by reducing or minimizing the training error, denoted as, $C\sum_{i=1}^{N}(\xi_i+\xi_i^*)$, also, it reduces the regularization term, represented as $(1/2)\omega^T\omega$. The above method is all followed up as per the principle of SRM., where both regularization term and training error are reduced or minimized simultaneously at the same time. In case of traditional or simple least square regression, ε is always zero and also datas are not mapped into higher dimensional spaces. Therefore, SVR/SVM is a more flexible way of solving out the regression problems.

Now, as we know that ϕ might mapped the X_i in a very high or infinite dimensional space, so this may lead to the problem of dimensionality, so in order to avoid the dimensionality problem, instead of solving ω for equation (35) in high dimension , the dual problem of equation (35) is solved. So, this dual problem is organized totally in terms of training data and then, it is to minimize or reduce dual variables of Lagrangian $L_d(\alpha,\alpha^*)$. So, now the dual equation will be :

$$\min_{\alpha,\alpha^*} L_d(\alpha,\alpha^*)=\frac{1}{2}(\alpha-\alpha^*)^T G(\alpha-\alpha^*)+\varepsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{N}y_i(\alpha_i-\alpha_i^*) \tag{36}$$

subject to constraints,

$$\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0$$

$$0\le\alpha_i\le C,\ i=1,2,...................,N$$

$$0\le\alpha_i^*\le C,\ i=1,2,3.................,N$$

equation 3 represents ideal quadratic optimization problem. α_i and α_i^* are the unknown Lagrange multipliers corresponding to ξ_i, ξ_i^* and the inner product $G_{ij}=\phi(X_i)^T\phi(X_i)$. As no training data can be on both sides of the ε- insensitive tube, either α_i or α_i^* will be non-zero. Now, for the data lying inside the tube , both multipliers value will be zero. Since, the $\phi(X)$ has too many elements, so it will be costly and hectic to calculate inner product i.e. G_{ij}. Hence, we will be using "Kernel Trick" to perform mapping implicitly. It means to deploy some special forms, which are inner products in higher space, still we can calculate them in original space. Few examples related to above said concept are :

Polynomial kernel, $\phi(X_i)^T \phi(X_j) = (\gamma X_1^T X_2 + c_0)^p$ and

Radial Basis Function (RBF) kernel, represented as $\phi(X_i)^T \phi(X_j) = e^{-(\|X_1 - X_2\|^2 / 2\sigma^2)}$

So, the above functions are inner products in the very high dimensional space but still can be calculated in original dimensional space.

Now, Learning outcomes in N Lagrange multiplier pairs are (α_i, α_i^*). After, learning, the number of free(non-zero) parameters α_i or α_i^* is equal to the number of support vectors SV's .

Now, after finding Lagrange multiplier vectors α and α^*, an optimal desired weights vector of the kernel expansion is found out as,

$$\omega_0 = \alpha^* - \alpha \tag{37}$$

and an optimal bias b_0 as

$$b_0 = \frac{1}{N} \sum_{i=1}^{N} (y_i - g_i) \tag{38}$$

where, g = G ω_0, and the matrix G is corresponding kernel matrix G(X_i , X_j).

The best non-linear regression hyper function is given by:

$$y = f(x, \omega) = G\omega + b \tag{39}$$

6.2.2 Genetic Algorithm

The GA is an algorithm of searching for optimization, perfectly based or relied on the mechanism of natural selection and genetics. This algorithm works on the assumption that the genuine or best solution lies in zone of space having solutions containing higher probability of genuine or most accurate solution, and these zones or regions can be find out or identified by random or robust inspection(sampling) of solution space.

The ability of GA is that it can search out a very large solution space efficiently at a very low costing price, because it applies probabilistic transition rules instead of deterministic rules. Also, GA is highly applicable to those problems, where a small change in input result in huge non-linear behavior of solution space. The GA involves following three process in stages as follows :

(a) Population booting/Population initialization

(b) Operators

(c) Chromosome assessment/ Chromosome evaluation

These are described as follows :

(a) Population booting **:**

In Genetic Algorithm (GA) the initial population is generated in a random procedure, but the main point is that this initial population must satisfy all precedence relations and also must comprises of valid sequences.

(b) Operators:

Now, operators consist of certain sub stages like: selection ; Reproduction ; Crossover ; Mutation ; Migration. they are explained as follows:

(i) Selection : Selecting the selection operator is a important step in GA algorithm. Researchers have recommended various selection operators, like for example: Tournament Selection model" given by Goldberg also similarly "Expected Value Model" and "Elitist model" given by De Jong.

(ii) Reproduction : It is another genetic operator. In Reproduction, basically the images of strings are pasted into a separately new string known as "Mating pool", as per their values of fitness. Therefore, it implies that the strings having higher value of fitness will be having greater chances of donating high counts of strings, as the search process progresses.

(iii) Crossover : The crossover operator main function is to examine progress of the search. Also, at the same time it exchange the parents string partially so as to give birth to offspring.

(iv) Mutation : The process such as crossover and reproduction creates several new chromosomes but still they failed to introduce or produce any new information at the bit level into the population. Mutation is a process in which information contained in chromosomes are occasionally and randomly oriented. Basically in mutation, genes mutate (alter or change) so as to become new genes having lower probability. Therefore so as to copy the evolution process, the mutation introduces bits to alter having lower probability, by inverting a bit of chromosome, which is selected randomly.

(v) Migration : It is defined as the individuals motion between the subgroups of the population. Quite often, the best performing individuals of one subpopulation replaces the worst performing individual of another subgroup of population. Parameters defining the mutation are : Interval, Fraction and Direction.

(c) Chromosome Evaluation / Chromosome Assessment :

In GA finally, the chromosome quality are checked and evaluated and the algorithm converge those genes into new off-springs, thereby resulting into new set of solution or new genes. Therefore, the algorithm terminates itself, if the algorithm fails to produce offspring, having property different from previous genes.

6.2.3 Genetic Algorithm(GA) based Support Vector Machine(SVM)

In case of SVM model, three free parameters are(C, ε, ϕ) C is Cost error ; ε is width of the tube ; ϕ is mapping function . The performance of the SVM models are greatly affected by these free parameters, so the main task for the researcher is how to select adequate parameters value, which will lead to genuine performance. Therefore, the performance of SVM models is directly dependent upon the adequate and appropriate values of parameters.

However, still there is a lack in structured methods for selecting parameters values. Also, it is unknown that at what combinational values of parameters does the SVM model performs best. So, optimizing the parameters for the SVM model is a necessary step for predicting the best performance. In this research work, the optimization technique of Genetic Algorithm (GA) is adopted in order to find out the free parameters values, which led to better performance of SVM model.

GA(genetic algorithm) is defined as a random search technique, which is directed in nature and it is mainly applicable to the optimization problems, where the analytical solution is very tough to obtain, just because of large number of parameters. GA is basically utilized to find out the optimal solution over a range globally.

The procedure flows as follows:

STEP -1 : First of all, SVM parameters are encoded and initialised, thus establishing an initial population of chromosomes i.e. By creating certain encoding programs, the initial population is created.

STEP -2 : Evaluating fitness

Now in this step, we evaluate fitness of trained SVM by taking each value of chromosome's gene as SVM parameters and then training them set wise as input and outputs sets respectively. After completing, the fitness value of chromosome should be evaluated and cross checked again.

In order to find out the fitness value of each chromosome, we use NRMSE as a evaluating parameter, here NRMSE is normalised root mean square error, as shown in eqn. 40.

$$fitness\ function = -\sqrt{\frac{\sum_{i=1}^{n}(a_i - f_i)^2}{\sum_{i=1}^{n} a_i^{\,2}}} \tag{40}$$

STEP -3 : Selection, Operation, Crossover and Mutation

Now out of the above mentioned steps, elaborating them in details : Selection is done in order to select the top performing chromosomes for reproducing. On basis of the fitness function, as calculated in eqn. 40. we can predict the chromosomes yielding out much better offspring in the upcoming batch by means of roulette wheel. Crossover is randomly done so as to swap genes among the two chromosomes. The operation of mutation creeps down the crossover operation, and here it find out that whether a chromosome can be altered or change in next generation or not. Now, after the above four operations, the offspring generated will be replacing the old batches and thus forms the new population in the upcoming generation by the same above four operation viz. Selection, Operation, Crossover and Mutation. This process of evolution proceeds on in same way, unless and until the termination condition are satisfied.

6.3 Results and Discussions

In this section the result of the simulation are presented and discussed through tabular comparative study of GA based SVM with respect to Linear regression, ANN/MLP and RNN. All the models are run for same number of epochs, and also having same number of input vector size, hidden layers, output vector size. Now below Fig. 6.6 shows the forecast vs actual plot for Linear regression and Fig. 6.7 shows the bar plot for the same. Also Fig. 6.8 shows the forecast vs actual plot for ANN/MLP and Fig. 6.9 shows the bar plot for same. While Fig. 6.10 shows the forecast vs actual plot for RNN and Fig. 6.11 shows the bar plot for same. Also, Fig. 6.12 shows the tracing graph for RNN. Similarly, Fig. 6.13 shows the forecast vs actual plot for GA based SVM and Fig. 6.14 shows the bar plot for same, and Fig. 6.15 shows the mape plot for GA based SVM.

The graphs or plot are made between y_true and y_pred values. Where, y_true are the True or actual value of the target/output while y_pred is the forecasted or predicted value of the target/output.

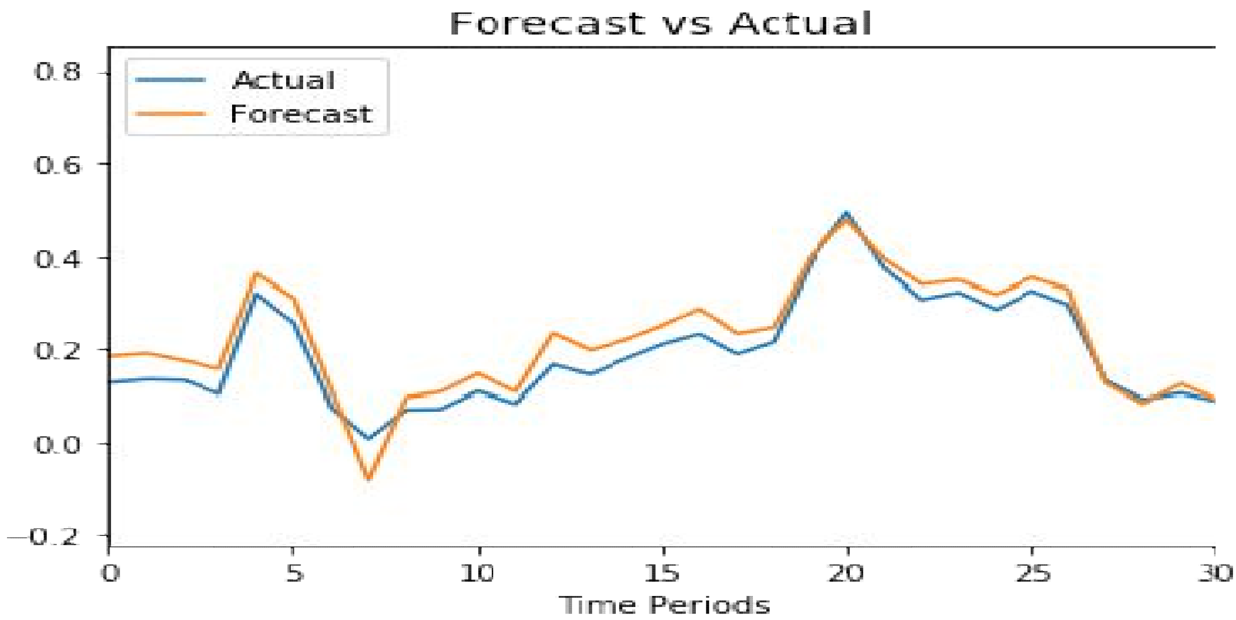

Figure. 6.6. Forecast vs Actual plot for Linear regression

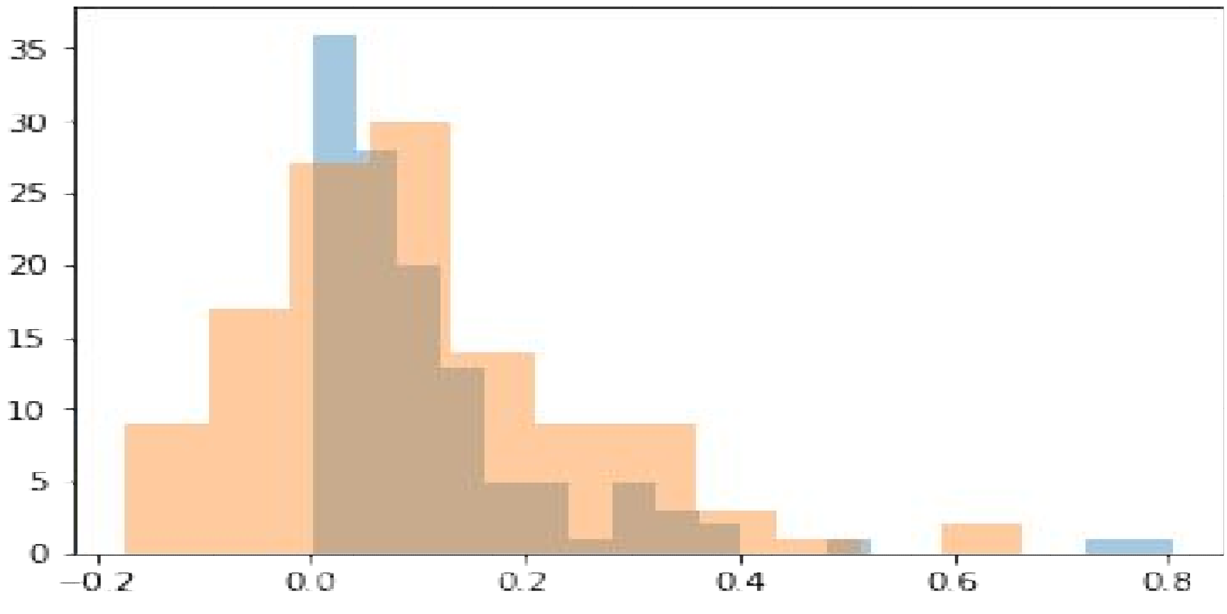

Figure. 6.7. Forecast vs Actual bar plot for Linear regression

Now, for ANN/MLP(multi layer perceptron) all the figures are shown below.

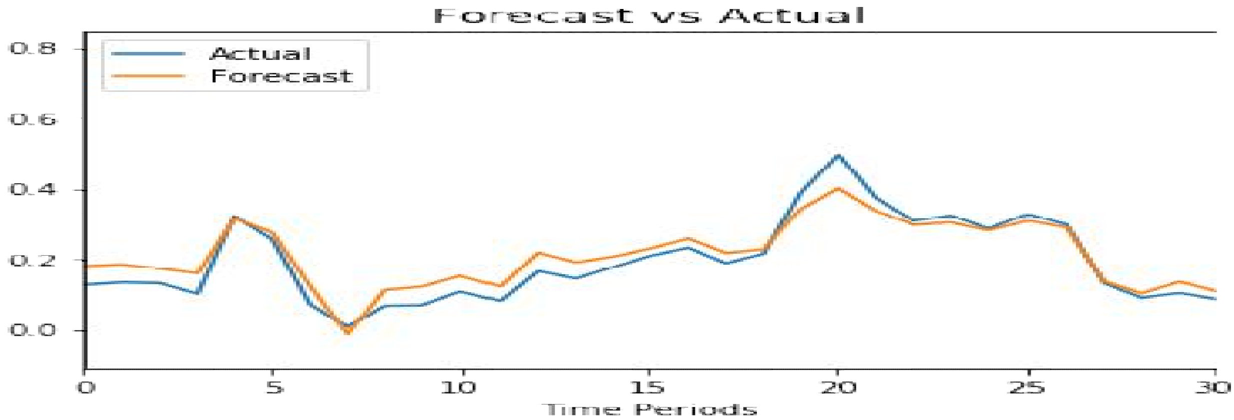

Figure. 6.8 Forecast vs Actual plot for ANN/MLP

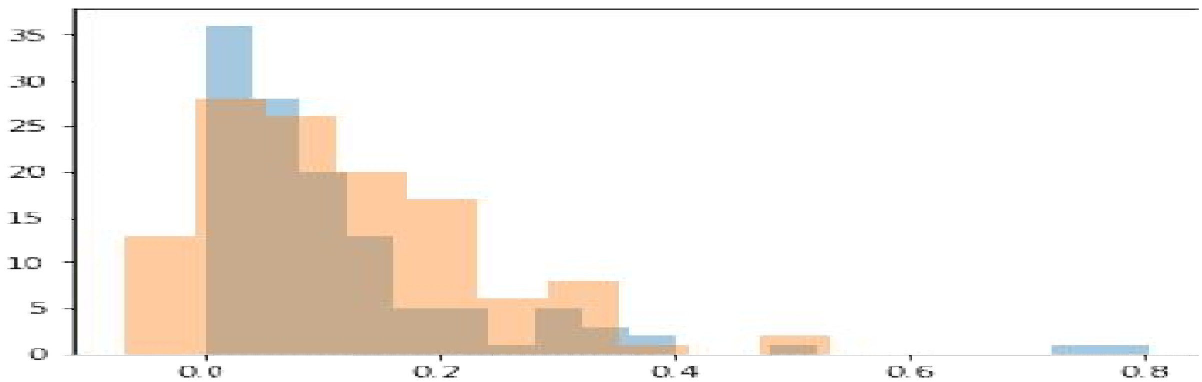

Figure. 6.9. Forecast vs Actual bar plot for ANN/MLP

For, RNN all the figures are shown below :

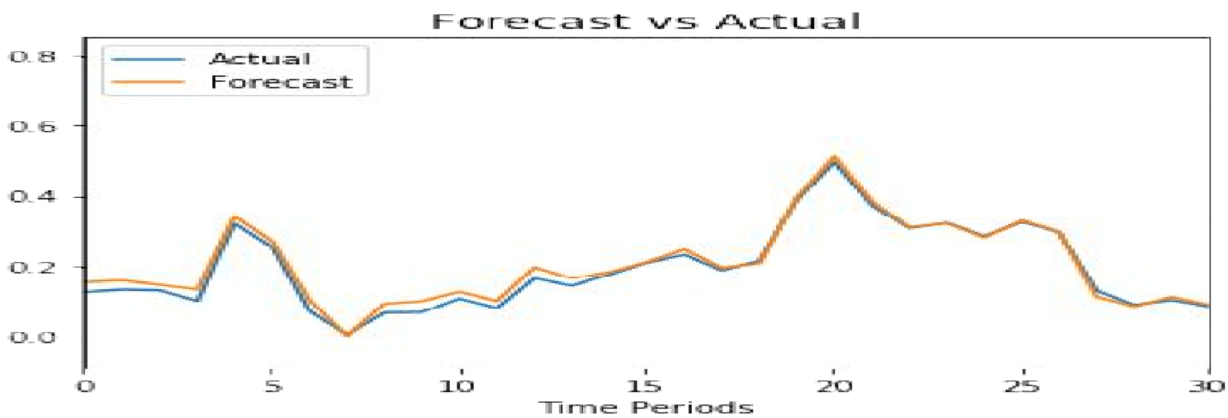

Figure.6.10. Forecast vs Actual plot for RNN

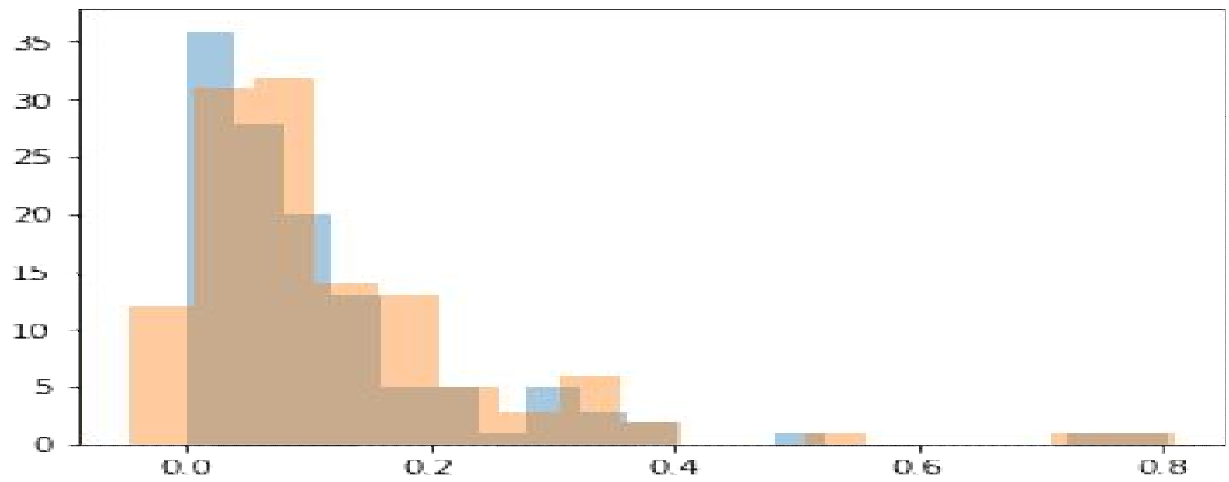

Figure. 6.11. Forecast vs Actual bar plot for RNN

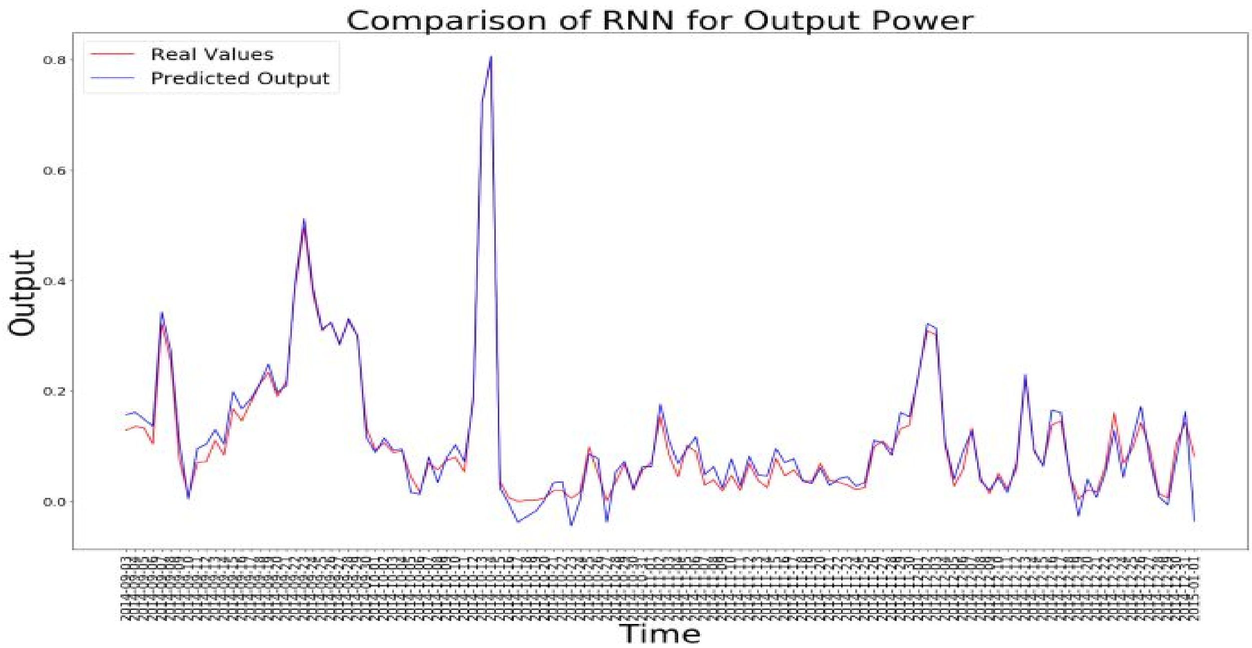

Figure. 6.12. Tracing graph for RNN

Now, for GA based SVM, all the graphical results are shown as follows :

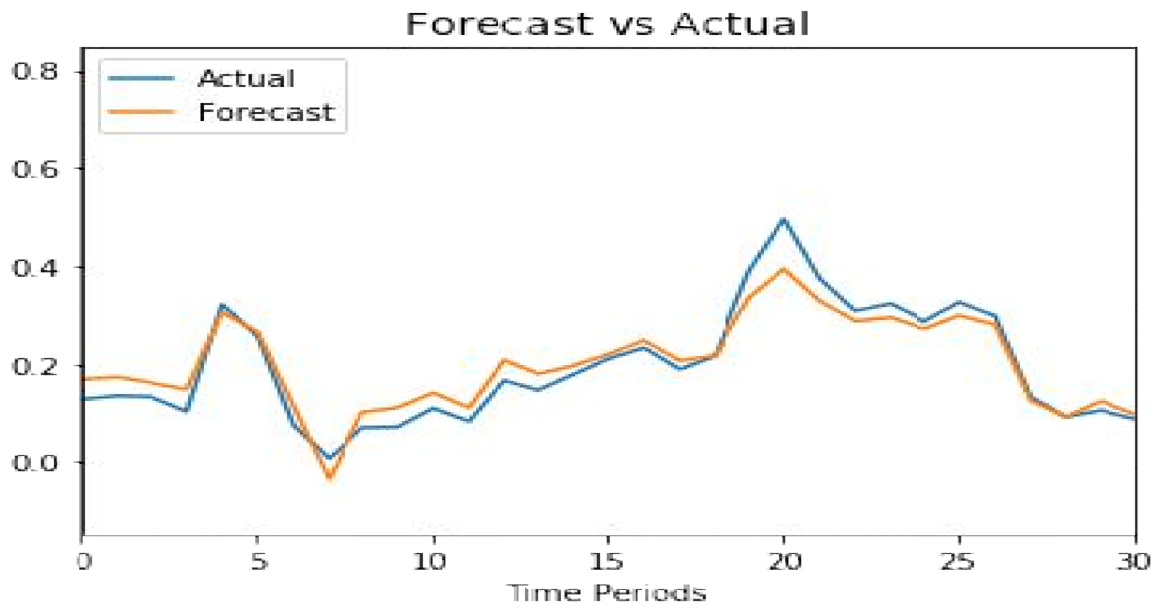

Figure. 6.13. Forecast vs Actual plot for GA based SVM

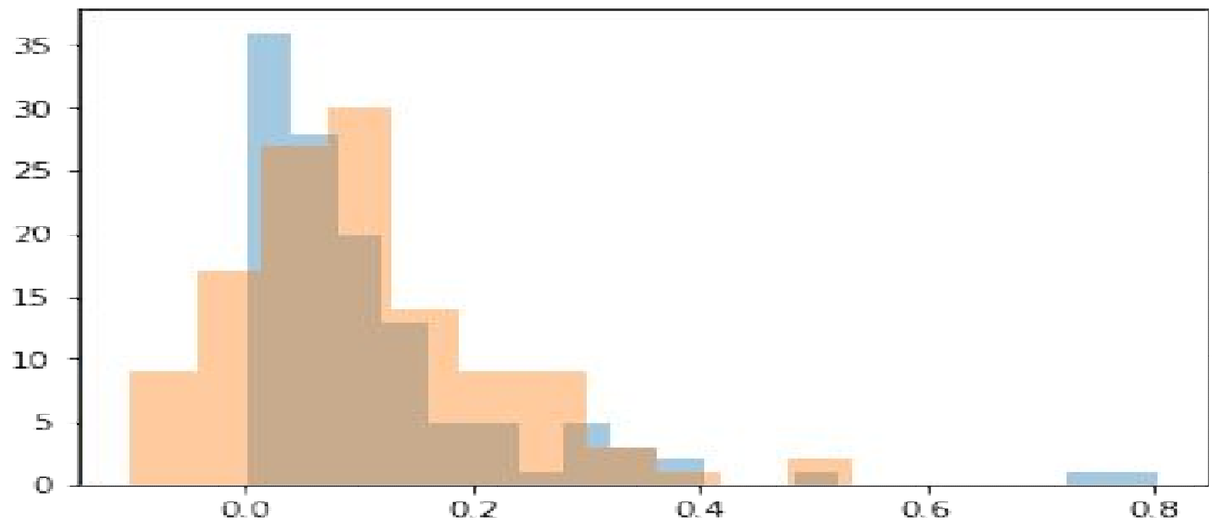

Figure. 6.14. Forecast vs Actual bar plot for GA based SVM

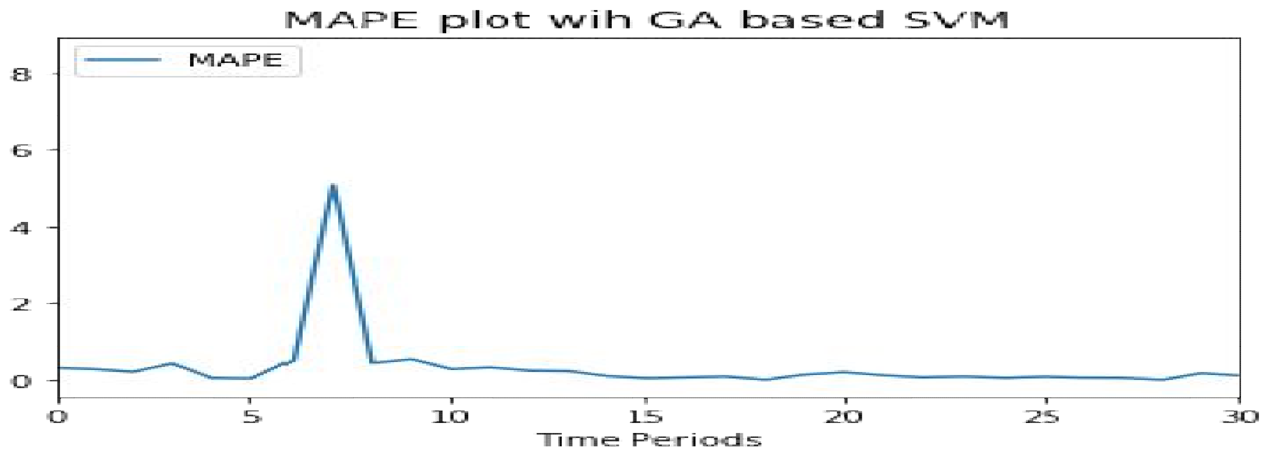

Figure. 6.15. MAPE plot with GA based SVM

Now, on the basis of performance parameters, all the algorithms are compared to GA based SVM, numerically and hence it is shown in the comparative table below.

Table 4. Comparative study of GA based SVM to other algorithm

PARAMETERS	LINEAR REGRESSION	ANN/MLP	RNN	GA based SVM
Mean Absolute Error (MAE)	0.047543	0.030737	0.015385	0.031417
Mean Absolute Percentage Error (MAPE)	5.355319	1.939639	1.148229	2.972836
Mean Square Error (MSE)	0.003917	0.002174	0.000426	0.002427
Root Mean Square Error (RMSE)	0.062586	0.046623	0.020640	0.049266
Variance	0.759744	0.866673	0.973868	0.851123

From above comparative table the observations are as follows :

(i) It is observed that the MAE value for GA based SVM is good as compared to Linear Regression and is comparable to ANN and RNN.

(ii) MAPE for GA based SVM is better than Linear Regression and is comparable to that of RNN and ANN.

(iii) MSE and RMSE values are least for RNN and almost same for ANN and GA based SVM.

Thus, it clearly proves from final outcomes that GA based SVM outperforms Linear Regression and is performing equally well as ANN/MLP, however it is performing lesser as compared to RNN.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion and Future Work :

Wind power prediction is very important for the scheduling of wind power on regular basis. In this thesis mainly three case studies are provided, along with some introductory and basics of the statistical tools which can be used for forecasting or predictive analysis. The case studies have following conclusions.

The first case study presents three different artificial intelligence networks NARX, NLIO and RNN to forecast the day ahead wind power generation in Kolkata region of India. RNN is found to be the best algorithm that can be used in case of short-term wind power forecasting. NARX model as well as NLIO model cannot be used for time series forecasting as they are having a very high value of error as compared to RNN algorithm. In future, hybrid models will be applied to further reduce the error in wind power forecasting.

In the second case study artificial intelligence models namely RNN, LSTM and GBM have been implemented to determine short term forecasting of wind power using data of Kolkata region of India. From the simulation results, it is evident that the LSTM, RNN and GBM , all are able to forecast the wind power with good accuracy but LSTM performs better than the RNN and GBM in the wind power forecasting. Further, it can be deduced that the future work could be based on the hybrid models of LSTM, GBM or RNN also other parameters such as wind direction and temperature may be taken as input to the network.

The third case study, GA based SVM algorithm have been implemented to determine short term daily average forecasting of wind power using data of Kolkata region of India. From the simulation results, it is evident that the GA based SVM outperforms the Linear Regression and is almost equally well as compared to ANN or Multi layer perceptron, however GA based SVM perform less as compared to RNN in the wind power forecasting. Further, it can be deduced that the future work could be based on the other algorithm to be used in feature selection process instead of GA. also other parameters such as wind direction and temperature may be taken as input to the network

REFERENCES :

[1] Adela Bâral, George Căruţaşu, Cornelia Botezatu, Alexandru Pîrjan: Comparative Analysis between wind and solar forecasting methods using Artificial Neural Networks, 16th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2015, pp. 89-94, Budapest 2016.

[2] Chao-Ming Huang , Chung-Jen Kuo , Yann-Chang Huang: Short-term wind power forecasting and uncertainty analysis using a hybrid intelligent method, IET Renewable Power Generation, 11(5), pp. 678-687 (2017).

[3] Murat Kayri :Predictive Abilities of Bayesian Regularization and Levenberg - Marquardt Algorithms in Artificial Neural Networks: A comparative Empirical study on social data ,Mathematical and Computational Applications (MDPI),pp 01-11 May 2016.

[4] Harvinder Kaur , Dalwinder Singh Salaria :Bayesian Regularization based neural network tool for software effort estimation , Global Journal of Computer Science and Technology neural & Artificial Intelligence , 0975-4350 , (2013).

[5] Omolaye O Philip , Badmos T Adeleke , "predictive and comparative Analysis of NARX and NIO timeSeies Prediction " , American Journal of Engineering Research (AJER) , pp. 2320-0936 (2017).

[6] Z.O Olaofe,K A Folly: Wind power estimation using Recurrent Neural Network Technique , IEEE, July (2012).

[7] Shilpa G.N. , Dr. G.S.Sheshadri: Short-term Load Forecasting using ARIMA Model for Karnataka State Electrical Load ,International Journal of Engineering Research and Development , 13(7), pp.75-79(2017).

[8] Shady Mahmoud Elgarhy , Mahmoud M. Othman , Adel Taha , Hany M. Hasanien : Short Term Load Forecasting Using ANN Technique , International Middle East Power System Conference (MEPCON), pp. 1385- 1394, Egypt (2017).

[9] M M Tripathi, SN Singh, KG Upadhyay: Price forecasting in competitive electricity markets: An analysis, Proceedings of International Conference on Energy Engineering, ICEE 2009, pp. 7-9, Puducherry (2009).

[10] Jaime Buitrago , Shihab Asfour : Short-Term Forecasting of Electric Loads using Nonlinear Autoregressive Artificial Neural Networks with Exogenous Vector Inputs , Energies(MDPI), Jan 2017, pp. 10-40 ,Miami (2017).

[11] A. V. Devadoss and T. A. A. Ligori, “Forecasting of stock prices using multi layer perceptron,” Int J Comput Algorithm, vol. 2, pp. 440–449, 2013.

[12] G. Batres-Estrada, “Deep learning for multivariate financial time series,” ser. Technical Report, Stockholm, May 2015.

[13] Paolo Burlando [a]RenzoRosso [a]Luis G.Cadavid[b]Jose D.Salas[b] : Forecasting of short-term rainfall using ARMA models , vol. 144, issue 1-4 , April 1993, pages 193-211.

[14] Adebile, Olukayode, Shangodoyin, Kehinde, Arnab Raghunath : Forecasting performance of logistic STAR model: an alternative version to the original LSTAR models , vol. 3 , - MASA. Model Assisted Statistics and Applications ,2018/12/16.

[15] Bollerslev, T. (2009). Glossary to ARCH (GARCH). Volatility and Time Series Econometrics: Essays in Honour of Robert F. Engle. Oxford. Oxford University press.

[16] Awartani, A. M. B. Corradi, V. (2005). Predicting the volatility of the S&P-500 stock index via GARCH models: the role of asymmetries. International Journal of Forecasting. 21, page 167-183.

[17] Nieto, Fabio. (2008). Forecasting with univariate TAR models. Statistical Methodology. 5. 263-276. 10.1016/j.stamet.2007.09.002.

[18] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction.” in IJCAI, 2015, pp. 2327–2333.

[19] A. Victor Devadoss , T. Antony Alphonnse Ligori : "Forecasting of Stock Prices Using Multi Layer Perceptron",vol 2,International Journal of Computing Algorithm , December 2013, pages 440-449.

[20] J. Roman and A. Jameel, “Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns,” in System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on,, vol. 2. IEEE, 1996, pp. 454–460.

[21] H. Jia, “Investigation into the effectiveness of long short term memory networks for stock price prediction,” arXiv preprint arXiv:1603.07893,2016.

[22] Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P, " STOCK PRICE PREDICTION USING LSTM,RNN AND CNN-SLIDING WINDOW MODEL" 978-1-5090-6367-3/17/$31.00 ©2017 IEEE , September 2017.

[23] Friedman, Jerome H, “Greedy Function Approximation: A Gradient Boosting Machine.” Annals of Statistics (2001): 1189-1232.

[24] Wei Sun, " Research on GA-SVM Model for Short Term Load Forecasting Based on LDM-PCA Technique ."Journal of Computational Information Systems 6:10(2010), pp. 3183-3189, October 2010.

[25] Han Xiaojuan , Zhang Xilin ,and Gao Bo, "Short-term Wind Speed Prediction Model of LS-SVM Based on Genetic Algorithm" , Y.Wu(Ed.): International Conference on WTCS 2009, AISC 116, pp. 221-229 (2009).

[26] L.M. Saini , S.K.Aggarwal , A.Kumar, "Parameter optimisation using genetic algorithm for support vector machine -based price forecasting model in National electricity market.."IET generation , Transmission , Distribution , Vol. 4, Iss. 1, pp. 36-49 (2010).

[27] Cristian - Dragos Dumitru , Adrian Gligor, "Daily Average Wind Energy Forecasting using Artificial Neural Networks" 10th International Conference Interdisciplinarity in Engineering , INTER -ENG 2016, pp. 1877-7058 © 2017 The Authors. Published by Elsevier Ltd.

[28] Thanasis G. Barbounis, John B. Theocharis, Minas C. Alexiadis, and Petros S. Dokopoulos,, “Long-TermWind Speed and Power Forecasting Using Local Recurrent Neural Network Models,” IEEE Transactions on Energy Conversion, vol. 21, no. 1, March 2006.

[29] GiacomoCapizzi, Christian Napoli, and Francesco Bonanno, “Innovative Second-Generation Wavelets Construction with Recurrent Neural Networks for Solar Radiation

Forecasting," IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 11, November 2012.

[30] Hans Bludszuweit, José Antonio Domínguez-Navarro and Andrés Llombart, "Statistical analysis of wind power forecast error," IEEE Transactions on Power Systems, vol. 23, no. 3, August 2008

[31] G .N.Kariniotakis, G. S. Stavrakakis and E.F.Nogaret, "wind power forecasting using advanced neural network models," IEEE Transaction on Energy Conversion ,vol 11.,no.4,December 1996.

[32] Cameron W. Potter, Michael Negnevitsky ,"very short - term wind forecasting for Tasmania power Generation," IEEE Transactions on Power Systems, vol. 21, no. 2, may 2006.

[33] T. S. Nielsen, H.MadsenJ, and J. Tøfting, "Experiences with statistical methods for wind power prediction," in Proc. Eur. Wind Energy Conf. (EWEC), 1999, pp. 1066–1069.

[34] T. S. Nielsen, H. MadsenJ, and J. Tøfting, WPPT, a tool for on-line wind power prediction International Energy Agency,Wind Forecasting Techniques Tech. Rep., 2000.

[35] NimaAmjady, Farshid Keynia and Hamidreza Zareipour, "wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization,"IEEE Transactions on Sustainable Energy, vol. 2, no. 3, July 2011.

[36] J. P. S. Catalão, H. M. I. Pousinho and V. M. F. Mendes, "hybrid wavelet-PSO-ANFIS approach for short-Term wind power forecasting in Portugal," IEEE Transactions on Sustainable Energy, vol. 2, no. 1, January2011.

[37] Saeed, Aqsa & Khan, Asifullah. (2018). Adaptive Transfer Learning in Deep Neural Networks: Wind Power Prediction using Knowledge Transfer from Region to Region and Between Different Task Domains.

[38] Stephen F. Witt, Christine A. Witt, Forecasting Tourism Demand : A review of empirical research, International Journal of Forecasting, Volume 11, Issue 3, September 1995, pages. 447-475.

[39] Victor Richmond R Jose, Robert L. Winkler , Simple Robust Averages of forecasts : some empirical results, Volume 4, Issue 1, Jan- March 2008, page 163-169.

[40] Charles C Holt, Forecasting seasonals and trends by exponentially weighted moving averages, International Journal of Forecasting Volume 20, Issue 1, Jan-March 2004, Page 5-10.

[41] Peter R Winters, Forecasting Sales by exponentially weighted moving averages, Management science journal, Vol. 6, issue 3, 1960.

[42] Howard Grubb, Alexina Mason, Long lead-time forecasting of UK air passengers by Holt-Winters method with damped trend, International Journal of Forecasting, Volume 17, issue 1, Jan - March 2001, page 71-82.
[43] Press Information Bureau ,Government of India, Ministry of New and Renewable Energy, Year End Review 2018- MNRE, 10-December 2018.