# FORECASTING OF RENEWABLE ENERGY AND LOAD USING SLIDING WINDOW AND NEURAL NETWORK APPROACH FOR MICROGRID

*A report on fulfilment submitted for the dissertation work*

*of*

**MASTERS OF ENGINEERING**

*in*

**POWER SYSTEMS**

*Submitted by*

Kangujam Anelis
(Registration No.: 801742014)

*Under the guidance of*

Mr. Jitender Kaushal
Lecturer, EIED
TIET
Patiala

Dr. Prasenjit Basak
Associate Professor, EIED
TIET
Patiala

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

**2019**

**Electrical and Instrumentation Engineering Department**
**Thapar Institute of Engineering & Technology, Patiala**
***(Declared as Deemed-to-be-University u/s 3 of the UGC Act., 1956)***
**Post Bag No. 32, Patiala – 147004**
**Punjab (India)**

# DECLARATION

I hereby certify that the work which is presented in dissertation entitled, "**FORECASTING OF RENEWABLE ENERGY AND LOAD USING SLIDING WINDOW AND NEURAL NETWORK APPROACH FOR MICROGRID**", in partial fulfillment of the requirements for the award of the degree of Master of Engineering in Power Systems, submitted to Electrical & Instrumentation Engineering Department of Thapar Institute of Engineering & Technology (Deemed to be University) is as authentic record of my own work carried under the supervision of Dr. Prasenjit Basak, Associate Professor, EIED and Mr. Jitender Kaushal, Lecturer, EIED. It refers others researcher's work which are duly listed in the reference section. The matter contained in this dissertation has not been submitted, neither in part nor in full to any other degree to any other university or institute except as reported in text and references.

Place: TIET, Patiala

Date: 20/09/2019

Kangujam Anelis

**Kangujam Anelis**

**Roll No.: 801742014**

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 20/09/2019

Mr. Jitender Kaushal
Lecturer, EIED,
TIET, Patiala

Dr. Prasenjit Basak
Associate Professor, EIED,
TIET, Patiala

# ACKNOWLEDGEMENT

A formal statement of acknowledgment will hardly meet the ends of justice in the matter of expressing my deep sense of gratitude and obligation to all those who helped us in the completion of M.E dissertation work. I would like to extend profound gratitude to my both mentors, Dr. Prasenjit Basak and Mr. Jitender Kaushal for guiding me whenever I came across any trouble or question in my mind. Without their help and guidance, it would never been possible to gain a lot of experience and knowledge.

I would like to thank Sr. Prof (Dr.) R.S. Kaler, Head, EIED, TIET; Dr. Nitin Narang (PG coordinator, EIED, TIET) and Ms. Manbir Kaur (former PG coordinator, EIED, TIET) for providing a healthy learning environment to explore as much we can.

Finally, I would like to thank my family and friends who have always been a motivation and support for me during the dissertation work.

Kangujam Anelis

(Registration No.: 801742014)

# TABLE OF CONTENTS


| | | | |
|---|---|---|---|
| **DECLARATION** | | | **i** |
| **ACKNOWLEDGEMENTS** | | | **ii** |
| **LIST OF TABLES** | | | **v** |
| **LIST OF FIGURES** | | | **vi** |
| **ABBREVIATIONS** | | | **viii** |
| **ABSTRACT** | | | **ix** |
| **CHAPTER - 1** | **INTRODUCTION** | | **1-3** |
| | 1.1 | Background | 1 |
| | 1.2 | Application of ANN in microgrid | 1 |
| | 1.3 | Importance of controller in microgrid | 3 |
| **CHAPTER - 2** | **LITERATURE SURVEY** | | **4-7** |
| | 2.1 | Gaps in research | 7 |
| | 2.2 | Objectives of study | 7 |
| **CHAPTER - 3** | **PROPOSED METHODOLOGY** | | **8-13** |
| | 3.1 | ANN model for forecasting | 8 |
| | 3.2 | Forecasting methodology using LM algorithm | 9 |
| | 3.3 | Sliding Window algorithm for forecasting | 12 |
| **CHAPTER - 4** | **RESULT AND DISCUSSION** | | **15-22** |
| | 4.1 | Next day load forecasting using ANN | 17 |
| | 4.2 | Next day solar forecasting using ANN | 18 |
| | 4.3 | Next day wind forecasting using ANN | 19 |
| | 4.4 | Forecasting using Sliding Window Technique | 21 |
| | 4.4 | Comparison of methods for forecasting | 22 |

| | | |
|---|---|---|
| **CHAPTER - 5 CONCLUSION AND FUTURE SCOPE** | | **23** |
| 5.1 | Conclusion | 23 |
| 5.2 | Future scope | 23 |
| **REFERENCES** | | **24** |
| **APPENDIX** | | **27** |
| **PLAGIARISM REPORT** | | **54** |

# LIST OF TABLES


| Table No. | Caption | Page No |
|---|---|---|
| Table 4.1 | Comparison of Forecasting methods | 22 |

# LIST OF FIGURES


| Figure No | Caption | Page No |
|---|---|---|
| Figure 3.1 | Block diagram of the proposed ANN model for load forecasting | 9 |
| Figure 3.2 | Block diagram of the proposed ANN model for Solar Irradiance forecasting | 9 |
| Figure 3.3 | Block diagram of the proposed ANN model for Wind speed forecasting | 9 |
| Figure 3.4 | Flowchart for forecasting using ANN | 12 |
| Figure 3.5 | Sliding Window technique for load forecasting | 14 |
| Figure 3.6 | Sliding Window technique for Solar Irradiance forecasting | 14 |
| Figure 3.7 | Sliding Window technique for Wind speed forecasting | 14 |
| Figure 4.1 | Load vs time for past 31 days | 15 |
| Figure 4.2 | Solar Irradiance vs time for past 31 days | 15 |
| Figure 4.3 | Windspeed vs time for past 31 days | 15 |
| Figure 4.4 | Relative Humidity vs time for past 31 days | 16 |
| Figure 4.5 | Temperature vs time for past 31 days | 16 |
| Figure 4.6 | GHI vs time for past 31 days | 16 |
| Figure 4.7 | The ANN model | 17 |
| Figure 4.8 | Regression plot | 17 |
| Figure 4.9 | Actual load vs forecasted load | 17 |
| Figure 4.10 | The ANN model for Solar Irradiance forecasting | 18 |
| Figure 4.11 | Regression plot for solar irradiance forecasting using ANN | 18 |
| Figure 4.12 | Actual Solar Irradiance vs Forecasted solar irradiance based on ANN | 19 |
| Figure 4.13 | An ANN model for Wind Speed forecasting | 20 |
| Figure 4.14 | The regression plot for Windspeed forecasting using ANN | 20 |
| Figure 4.15 | The actual windspeed vs Forecasted Wind speed | 20 |

| | | |
|---|---|---|
| Figure 4.16 | The actual load vs forecasted load based on Sliding Window Approach | 21 |
| Figure 4.17 | The actual windspeed vs forecasted windspeed based on Sliding Window Approach | 21 |
| Figure 4.18 | The actual solar irradiance vs forecasted solar irradiance based on Sliding Window Approach | 21 |

# ABBREVIATIONS

| | |
|---|---|
| SWA | Sliding Window Approach |
| ANN | Artificial Neural Network |
| LM | Levenberg Marquart |
| MSE | Mean Squared Error |
| GHI | Global Horizontal Irradiance |
| NFTOOL | Neural Fitting Tool |

# ABSTRACT

This dissertation presents the study of application of Sliding Window Approach for forecasting. The past data can be utilized for predicting the future data. The data from 18th to 31st January and from 25th to 31st of January have been considered to forecast the data on 1st Feb. The current year's variation throughout the week is being matched with that of the previous year by using the mean of Sliding Window Approach and the best window is selected for forecasting. The selected window and the current year's weekly variations are used for the purpose of forecasting. The first objective of the work is to study the application of Sliding Window Approach for forecasting and the second objective is to propose a Sliding Window based algorithm for forecasting of data of Patiala in India using Matlab. The third objective is to compare the method of forecasting. The result for both the methods is compared and it is found satisfactory.

**Keywords:** Sliding Window Approach (SWA), Artificial Neural Network (ANN), Mean Squared Error (MSE), Forecasting, Microgrid.

# CHAPTER 1
# INTRODUCTION

## 1.1 BACKGROUND

An Artificial neural network (ANN) has a vast application in power system. An ANN has been implemented as a time series forecaster. Sliding Window approach (SWA) is popular in various time series applications such as in the medical, weather and financial domains. It is a temporal approximation over the actual values of the time series data. It identifies the segments of the first value of the time series and merges them into the next value until a specific error criterion is satisfied. After selecting the first segment, the next segment is selected from the end of the first segment. The process is repeated until all-time series data are segmented. In the time series segmentation, window size plays an important role in the segmented time series subsequence. For each window, w indicates the number of values being grouped into one window. Sliding Window Approach (SWA) is one of the most important approaches that are being used in many efficient time series analysis and is appropriate for various applications. The main functions of the SWA are to find the number of intervals and the width of the intervals. Large scale data can be handled by segmentation with SWA. This method is attractive because the segmentation can be implemented easily using an online algorithm.

## 1.2 APPLICATION OF ANN IN MICROGRID

ANN has a vast application in the field of microgrid. It can be implemented for the various applications in Microgrid. It is shown below:

### 1.2.1 FORECASTING

**Demand Forecasting**: The objective of demand forecasts is to forecast the demand in the power system. The load nature of microgrid is not uniform in nature. Due to its nature, there is accuracy problem in predicting demand. Some of the factors to be considered while developing the ANN model are

- The past data are given as a different element instead of clustering them into a single input
- It depends on the type of microgrid of different input parameter.

**Renewable Generation Forecasting**: An accurate prediction is needed in a Microgrid for the proper function. In microgrid, most of the generation sources are renewable.

**Wind Power Forecasting**: There are two types of wind power forecasting. The first type of wind power forecast, the wind speed is forecasted and converted into power. The second type predicts wind power. The difference between physical models and the statistical model is that one requires mathematical modelling of various energy sources is a complex task and another does not.

**Solar Power Forecasting**: Solar Power Forecasting: For the prediction of solar power, it depends on different input parameters such as solar radiation, temperature, pressure, relative humidity etc. Each model forecasts a solar power output and all the outputs can be combined recursively to deliver an accurate final forecast.

### 1.2.2 PROTECTION

Conventional techniques are faulty in nature. These can be overcome by the application of ANN. The ANN is a better option for designing an effective Microgrid system. The reasons are as follow:

- ANN take the whole input space in protection systems into account whereas conventional fault classification algorithms do not.
- ANN is best for solving pattern recognition problem.
- ANN are fault tolerating in nature.

.

### 1.2.3 INTELLIGENT DIAGONOSIS OF EQUIPMENT IN MICROGRID

All electrical equipments are subject to incipient faults. By utilizing diagnostic techniques, the incipient faults can be identified. The ANN is a better option for detecting faults in power system for mentioned reasons.

- They can interpolate from previous learnings.
- They are fault tolerant.
- They have non-linear function approximators.
- They are more suitable for extracting the relationship between input and output in fault diagnosis application.

Different approaches have been adopted by many researchers. An ANN approach has been adopted in [1] and [2] for the short-term load forecasting and solar energy generation prediction respectively. Based on Discrete Wavelet Transform and BP neural network, load forecasting has been done in [3]. Fuzzy Rule base approach has been adopted in [4] for protection of microgrids. For the optimal sizing of Battery energy storage system, ANN based feed forward neural network has been implemented in [5] and [6]. In [7], the authors have implemented a Deep neural network (DNN) based approach for analyzing the economic load dispatch problem in microgrid. For diagnosing and locating the fault in microgrid, the authors in [8] and [9] have adopted a multilayer feed forward neural network.

### 1.3 IMPORTANCE OF CONTROLLER IN MICROGRID

Microgrid central controller (MCC) plays an important role in control and operation of microgrid in both Grid connected mode and Islanded mode of operation. It compares the total generation with the load demand. In case of scenario where the demand in the system is greater than the power generation then non-critical loads will be shaded. The parameters such as frequency and voltage are both controlled by microgrid controller to maintain the stability of the system. For the purpose of proper coordination of the energy sources, microgrid controller is required. Microgrid central controller make the system more economical by reducing the operational cost to customers. Microgrid controller improved the power quality and stability of the system. ANN has become a great area of research.

In this report, the survey of the application of artificial neural network in microgrid has been done. The details of the literature survey which has been done on the application of ANN in microgrid and the issues in microgrid have been discussed in section II. The gap in research and the objective are discussed in this section. The proposed methodology has been discussed in section III. The system description is discussed in this section. The different method of forecasting by Sliding Window Approach and Neural network has been discussed in this section. Also, the algorithm associated the proposed methodology has been discussed. The result based on Sliding Window Approach and Neural network have been discussed in section IV. The comparison of different methods for forecasting are done in this section. The conclusion and future scope are discussed in section V. The references have been provided in section VI. The sliding window program for forecasting the data has been described in the appendix in last section.

# CHAPTER 2
# LITERATURE SURVEY, MATERIALS AND METHODS

In [1], the prediction of the stock prices of Taiwan construction companies one step ahead implementing the sliding-window metaheuristic optimization has been discussed. In stock market, the stock price is non-linear in nature so the hybrid model was proposed so that the efficiency of forecast would be improved. It was observed that the performance had been greatly improved by using this hybrid model for Taiwan system. The proposed model had been tested in Matlab environment. Even though it improved the performance, it has few drawbacks such as slow computational speed because of the complexity of the system and it has poor performance for long-term investment. Same approach could also be used for other growing markets in different parts of the world.

In [2], an Artificial Neural Network's application in the area of renewable energy has been presented. Generally, in the field of renewable energy forecasting, the multilayer perceptron network is used. A methodology is developed for forecasting the global horizontal irradiation.

In [3], a new method comprising of both neural network and sliding window technique for forecasting of power has been proposed. The performance in neural network is measured by the value of MSE obtained after forecasting. Since a small value of MSE is obtained, it indicates that it has a good performance. In the future, it can be extended for improving the performance of hybrid electric vehicle.

A data classification approach based on slide window neural networks has been proposed in [4]. It can be seen that a vast improvement in algorithm's property is there when the slide-window approach is followed. So, this makes the classification method based on this algorithm much better than the other conventional classification method.

In [5], a case study of regional basis in Turkey for forecasting natural gas demand ahead using optimized artificial bee colony-based artificial neural networks with Sliding Window Technique has been done. In this, only the past data of consumption is considered for predicting the day-ahead consumption data. The proposed model gave a better result than Back propagation training.

In [6], a method implemented to predict PV output power is discussed. The current data of weather condition and output power of the PV system were used and it result was compared with Meinel model.

In [7], the solar irradiation is predicted using back propagation algorithm. Three cases are conducted and tested. The best performance is obtained when the window size is three.

In [8], a sliding window is used for predicting the load. In this paper a sliding window method is adopted which performs well particularly for days where there are high levels of generation.

In [9], sliding window technique has been employed for prediction of wind velocity. From the analysis result, it could be observed that the MLP network with a 4-7-13-1 architecture is superior to others.

In [10], the sliding window approach with Early Stopping solves the problems for using the Levenberg–Marquardt algorithm on-line.

In [11], a named entity recognition algorithm based on sliding window to obtain and classify structure in natural language text has been discussed. The results showed the better accuracy as compared with other approaches

In [12], an Artificial Neural Network (ANN) technique: Radial Basis Function Network (RBFN) for data prediction using the concept of sliding window had been discussed. The historical data of earlier days was calculated by Weighted Moving Average (WMA). The performance of the model was checked using different accuracy measures like MAPE, MAD, MSE and RMSE. The results were found to be satisfactory.

In [13], a novel based on time series prediction and neural networks had been discussed. It can be concluded that the window size plays an important role in a neural network for prediction.

In [14], the modelling and the tracking of the crack spread has been performed based on a sliding window approach. In this, the developed model is a integration of particle swarm optimizer and a radial basis function neural network. It can be observed from the simulation result that it has forecasted with high accuracy.

In [15], a sliding-window time series analysis (SWTS) for San Francisco urban roads was done. It could be concluded from the result that the SWTS allows to predict the future vehicle speed data up to a predefined forecasting horizon.

In [16], a weather forecasting is done using sliding window algorithm. The accuracy can be enhanced by increasing the number of window size. The result can be further improved if neural network is incorporated with Sliding window approach.

The authors in [17] introduced the Sliding Window Algorithm (SWA) is better than the scatter search algorithm and bionomic algorithm.

In [18], ANN had been proposed for forecasting the solar energy generation for the control of microgrid. The irradiation, time and season were given as input to the system and the forecasted irradiation was obtained as an output. The datasets of 146 were given to the network for training the system.10 hidden neurons were implemented. The Levenberg-Marquardt algorithm was adopted for training the data. High accuracy in prediction of irradiation helps the microgrid to control effectively. Accurately predicting the solar energy generation is essential for the utility companies for planning. For the effective controllability of the system, balancing the load with the power generated is the main concern. The accuracy of the proposed ANN network was analyzed by using the mean square error. The mean square error of 0.5-9% was observed between the actual energy produced and the predicted power. The regression value was found to be 0.98981 which implied that there was a close relationship between the output and the target values.

In [19], the simulation of photovoltaic, solid oxide fuel cell and battery energy storage system based microgrid had been presented. The proposed model was simulated and it was compared with other method.

In [20], the PV/Fuel cell-based hybrid microgrid with help of simulation approach to improve the power quality in microgrid with ANN control had been discussed. The hybrid micro gird is growing due to the potential benefits in providing safe, reliable and sustainable electricity from renewable energy sources. The artificial neural network control was used for to improve the power quality in microgrid. A small hybrid microgrid had been modelled and simulated using the Simulink in the MATLAB. The simulation results were compared with Pi control.

In [21], a novel neuro-fuzzy controller based on data analytics was proposed in this paper to ensure stable operation of DEG-PV hybrid AC microgrid in both modes. The good performance of the proposed controller was obtained and it was tested under several conditions.

In [22], the improvement of the voltage profile of Microgrid using the neural network algorithm had been discussed. The Back-propagation algorithm was implemented and it could be concluded from the result that it is best for enhancing the microgrid’s voltage profile.

In [23], the PI Controller Research of UPQC in Micro-grid Based on RBF Neural Network had been addressed here. The desired value of input and output data are taken for the purpose of prediction in the proposed system. This approach has a vast application in microgrid.

In [24], Fuzzy-Neural Controller Design for Stability Enhancement of Microgrid. Fuzzy -neural-network approach was mostly suitable for the system that is nonlinear in nature. A Multilayer fuzzy network of five layers was implemented in the system. It had two input nodes and one output node. A bell-shaped function used as a membership function. The transient and steady state stability had been greatly improved in this approach as compared to conventional method controller. It was more economical in nature since there was no requirement of load shedding strategy and storage device.

Various techniques of forecasting have been discussed in the literature survey with different characteristics of input parameters for different system.

## 2.1 GAP IN RESEARCH

After comprehensive literature review on the application of Sliding Window Approach, it could be concluded that if other technique such as artificial neural network is incorporated with the Sliding Window Technique, the errors could be minimized further and the performance of the model would be improved.

## 2.2 OBJECTIVE OF THE WORK

1. To study Sliding Window Approach for forecasting of load and other renewable energy sources.
2. To propose a Sliding Window based algorithm for forecasting of load of Patiala Grid and other renewable sources in India using Matlab.
3. Comparison of Sliding Window Approach and Neural Network Technique for forecasting of load and other renewables sources.

# CHAPTER 3

# PROPOSED METHODOLOGY

The Sliding Window based algorithm has been proposed for forecasting the load, Solar irradiance and wind speed of Patiala Grid in India using Matlab**.** To predict the future data, the data in the previous week and in the past year is taken into account for the case of Sliding Window Approach. In case of ANN, the past data of 31 days are taken in order to forecast the future data.

## 3.1 ANN MODEL FOR FORECASTING

In this dissertation, an ANN model has been presented for the forecasting the next day load profile of Patiala Grid, next day Solar irradiance and next day windspeed. A two-layer feed-forward network is implemented here. The proper selection of an appropriate architecture is a key factor in designing the system. The sigmoid and linear activation functions have been employed in hidden layer and output layer respectively. The block diagram of the proposed ANN model is shown in Fig. 3.1. It has a total of 3 input parameters and 1 output parameter. The inputs to the neural network are temperature, relative humidity data and load profile of the previous day. In Fig. 3.2, a total of 3 input parameters and 1 output parameter are shown. The input to the neural network is temperature, GHI and previous solar irradiance. The output is the forecasted dollar irradiance. In Fig. 3.3, the input parameters are temperature, pressure and previous wind speed. The output parameter is the forecasted wind speed. The increase in the number of hidden layers affects the complexity of the Neural Network architecture and its performance. The ANN model functions in such a manner that the information is transferred in a feed forward manner from input to output layer via hidden layer and the weight is updated through back propagation. The well-known Levenberg-Marquart algorithm is applied to the ANN model. It adjusts the corresponding weight and bias and updates its value**.**

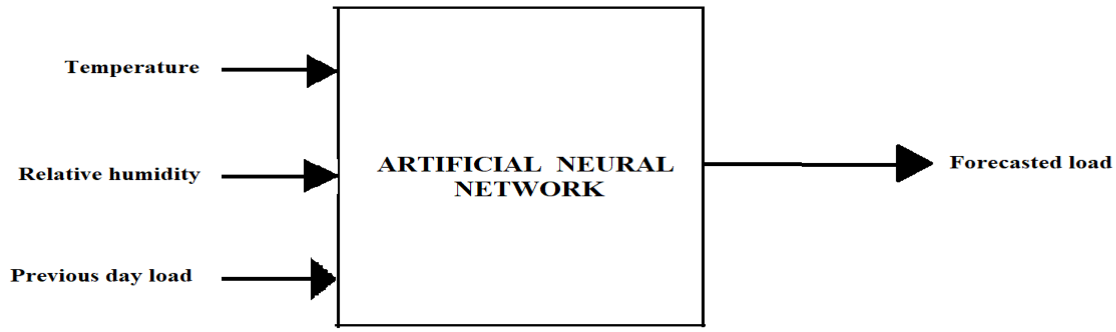



Fig.3.1. Block diagram of the proposed ANN model for load forecasting

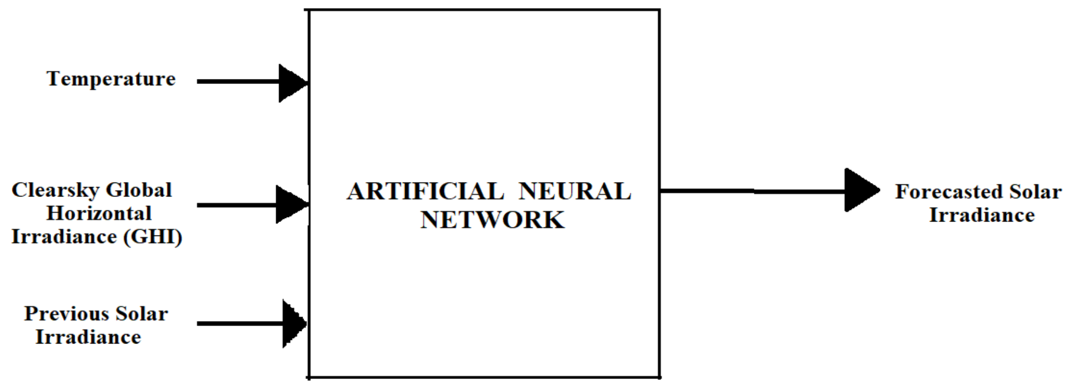



Fig.3. 2. Block diagram of the proposed ANN model for Solar Irradiance forecasting

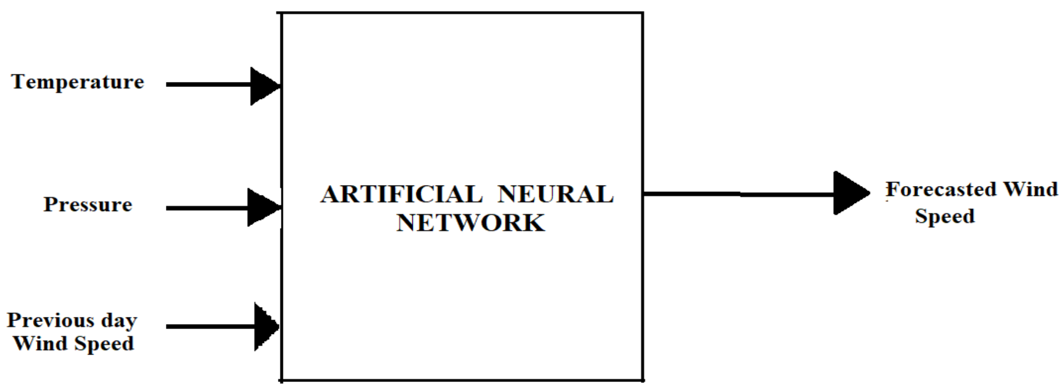



Fig. 3.3. Block diagram of the proposed ANN model for Wind speed forecasting

### 3.2 FORECASTING METHODOLOGY USING LM ALGORITHM

In this proposed dissertation work, data from 1st to 31st of January are implemented for testing and training the neural network. The whole data is been classified into three categories: 70% for training data, 15% for validation data and 15% for testing data. In these ANN models, an input layer comprises of 4 and 3 neurons and output layer is composed of 1 neuron for the different ANN models. In the hidden layer, 4 and 3 neurons are considered for the different ANN models as the performance of the proposed model is best. The number of the hidden neuron depends on the application and type of the ANN model.

The percentage error is given by

$$\%E = \frac{|Actual\ value - Forecasted\ value|}{Actual\ value} \times 100 \quad (1)$$

And

$$MSE = \frac{\sum(Actual\ value - Forecasted\ value)^2}{n} \quad (2)$$

Where n is equal to number of samples.

For LM optimization, the function to be optimized can be written as:

$$F(w)=\sum_{p}^{p=1}\left[\sum_{k=1}^{k}\left(d_{kp}-o_{kp}\right)^2\right] \quad (3)$$

Where w represents all the weights of the network, p represents the number of the patterns and k represents the number of outputs, $d_{kp}$ represents the desired value of the kth output and the $p^{th}$ pattern, $o_{kp}$ represents the actual value of the $k^{th}$ output and the $p^{th}$ pattern and N represents the number of weights. LM meld the speed of Gauss-newton's method with the stability of error back propagation algorithm during training. When $\mu$ is large, the learning process will follow error back propagation algorithm. When μ is small, the learning process will follow the Gauss–newton rule.

$$\Delta W = (J^t J + \mu I)^{-1} J^t e \quad (4)$$

The Jacobian matrix is defined as

$$J=\begin{pmatrix} \frac{\partial F(x_1,W)}{\partial w_1} & \cdots & \frac{\partial F(x_1,W)}{\partial w_w} \\ \vdots & \ddots & \vdots \\ \frac{\partial F(x_N,W)}{\partial w_1} & \cdots & \frac{\partial F(x_N,W)}{\partial w_w} \end{pmatrix} \quad (5)$$

The dimension of the above Jacobian matrix is N x W.

The steps involved in the LM are discussed below:

1. Evaluate the Jacobian matrix.

2. Evaluate the error gradient.

$$g = J^t E$$

3. Approximate the Hessian matrix using the cross-product Jacobian.

$$H = J^t J$$

4. Solve $(H+\lambda I)\delta = g$, to find $\delta$.

5. Update the weight of the network, w using $\delta$.

6. Reevaluate the sum of the error using updated weights.

7. If the sum of squared error has not decreased, discard the weights, increase $\lambda$ using v and go to step4.

8. Else decrease $\lambda$ using v and stop.

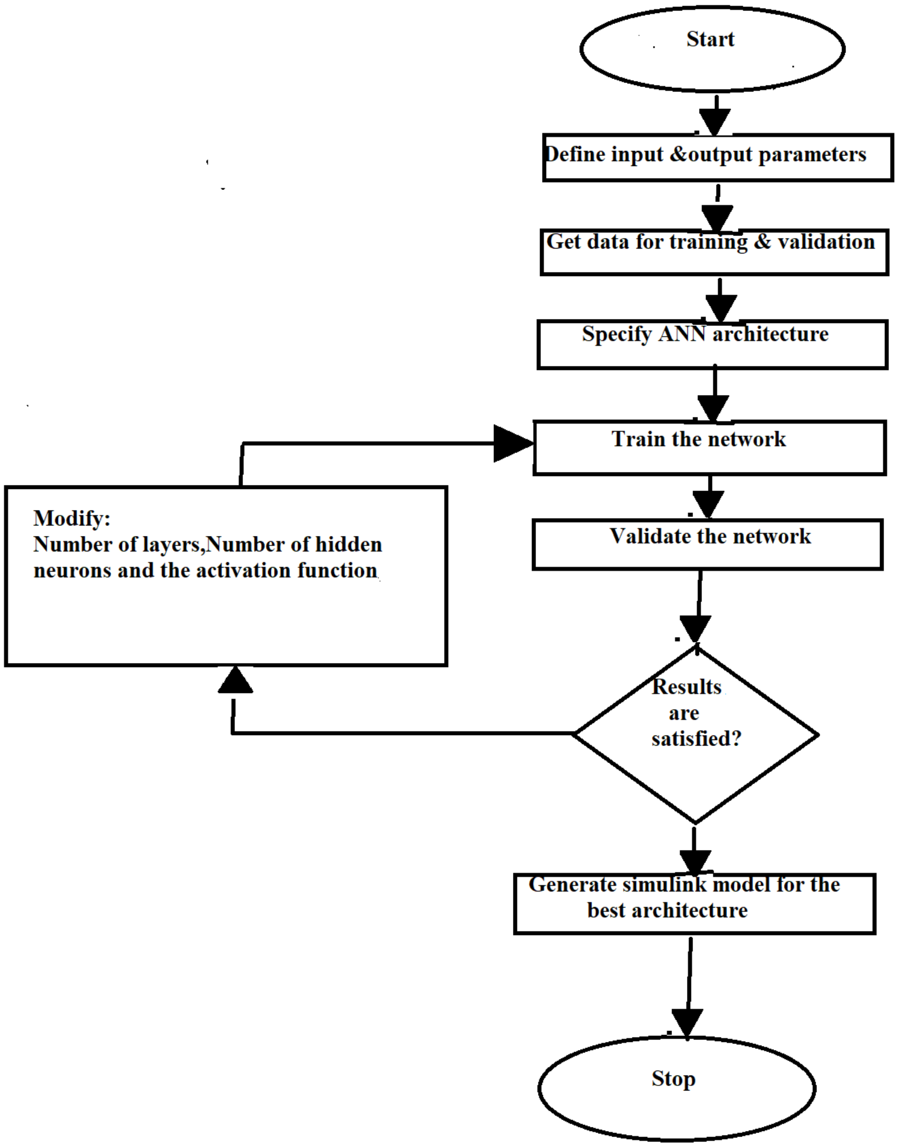



Fig. 3.4. Flowchart for forecasting using ANN

The flowchart used for forecasting the load, solar irradiance and wind speed is shown above.

### 3.3 SLIDING WINDOW ALGORITHM FOR FORECASTING

For the load forecasting of 1$^{st}$ Feb, the load data from 18$^{th}$ to 31$^{st}$ January of 2016 and the load data from 25$^{th}$ to 31$^{st}$ January of 2017 are taken into consideration. The same technique has been repeated for forecasting the future data for 31 days. In Sliding Window Algorithm, the selected window is obtained by comparing the variation of the present year for each day of the week with that of the previous year by using implementing the Sliding Window Technique. The selected window and the current year's weekly variations are used for the purpose of forecasting. The past year matrix 14x3 is divided into sliding windows. The Sliding Window of size of 7x3 is considered. There are in total of 8 sliding window. The Sliding Window concept is shown in Fig. 3.5, Fig. 3.6 and Fig. 3.7.

The steps in Sliding Window Algorithm is described below:

**Step1:** Consider a matrix "A" of last week for the current year's data of size 7x3.

**Step2:** Consider a matrix "B" of last two weeks for the previous year's data of size 14x3.

**Step3:** Make a sliding window of size 7x4 each from the matrix "B" as W1, W2, W3, W4, W5, W6, W7 and W8.

**Step 4:** For each sliding window, evaluate the Euclidean distance with the matrix "A" as ED1, ED2, ED3, ED4, ED5, ED6, ED7 and ED8.

**Step 5:** Choose the sliding window matrix which has the minimum ED.

**Step 6**: For k=1 to n

1. For $WC_k$, evaluate the variation vector for the matrix "A" of size 6x1 as "VA".
2. For $WC_k$, evaluate the variation vector for the matrix "B" of size 6x1 as "VB".
3. Mean1= Mean(VA).
4. Mean2= Mean(VB).
5. Predicted variation "VV" = (Mean1+Mean2)/2
6. Increase the previous day's load in consideration by" VV" in order to obtain the forecasted data.

**Step 7:** End.

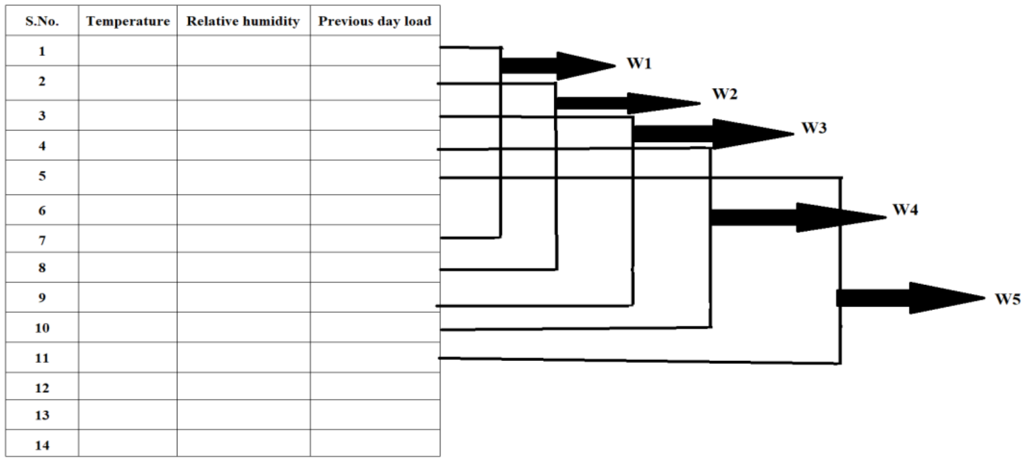



Fig. 3.5. Sliding Window technique for load forecasting

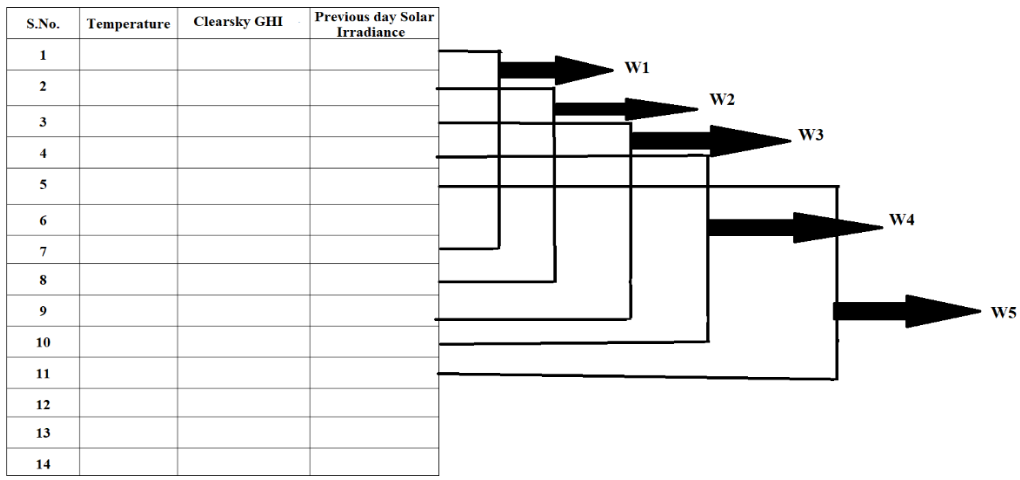



Fig. 3.6. Sliding Window technique for Solar Irradiance forecasting

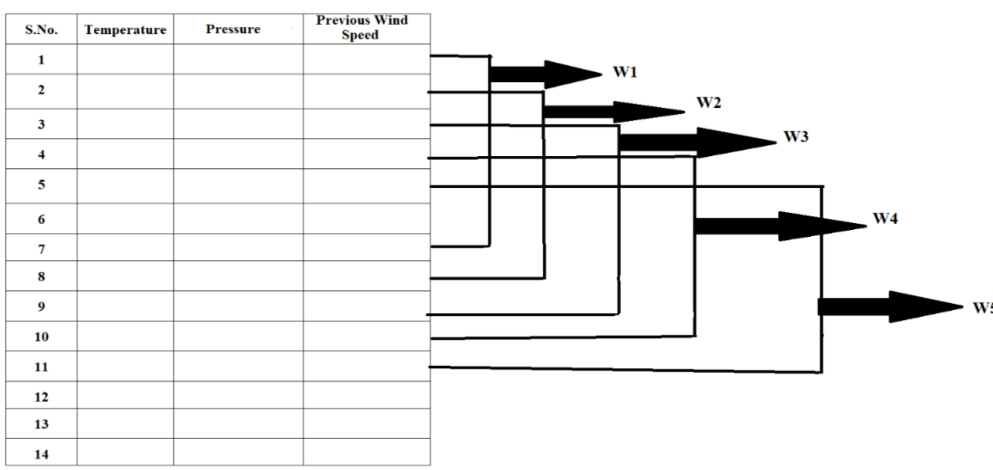



Fig. 3.7. Sliding Window technique for Wind speed forecasting

# CHAPTER 4

# RESULT AND DISCUSSION

In this section, the details of the next day forecasting and its case study have been described. Fig. 4.1 illustrates the nature of the load curve vs time for the past 31$^{st}$ days from 1$^{st}$ of January to 31$^{st}$ of January. Fig. 4.2 illustrates the nature of solar irradiance for the month of January. It can be observed that the solar irradiance has been gradually increasing due to the changes in weather condition.

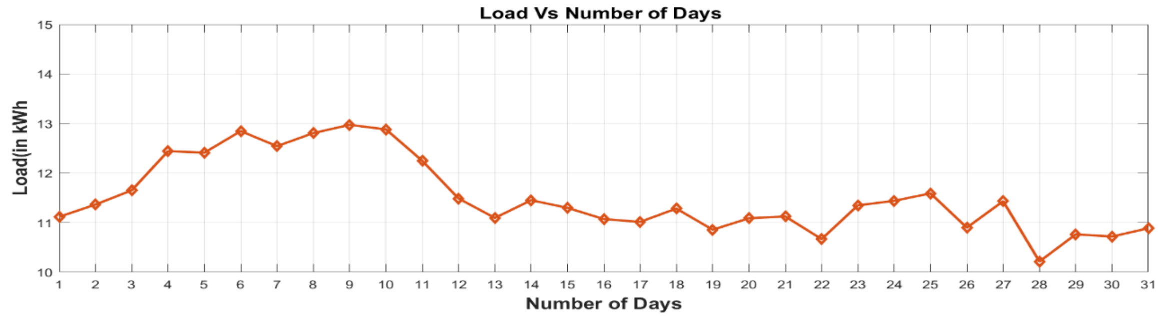



Fig. 4.1. Load vs time for past 31 days

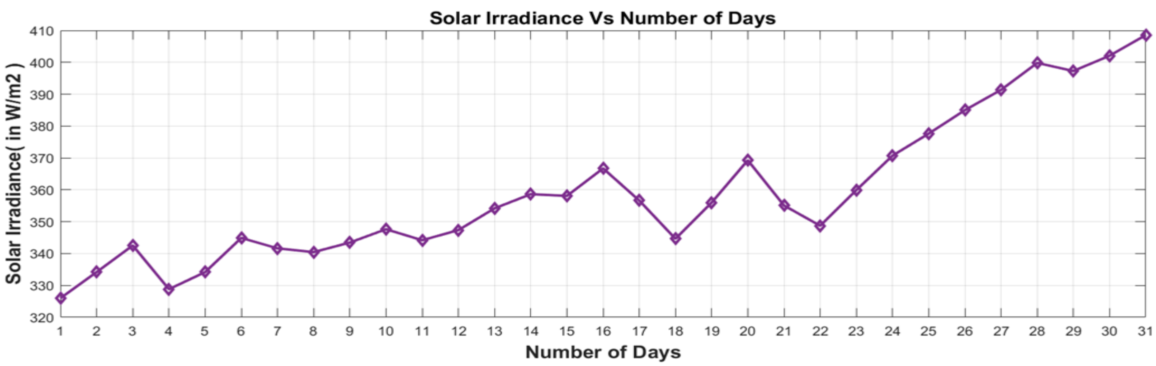



Fig. 4.2. Solar Irradiance vs time for past 31 days

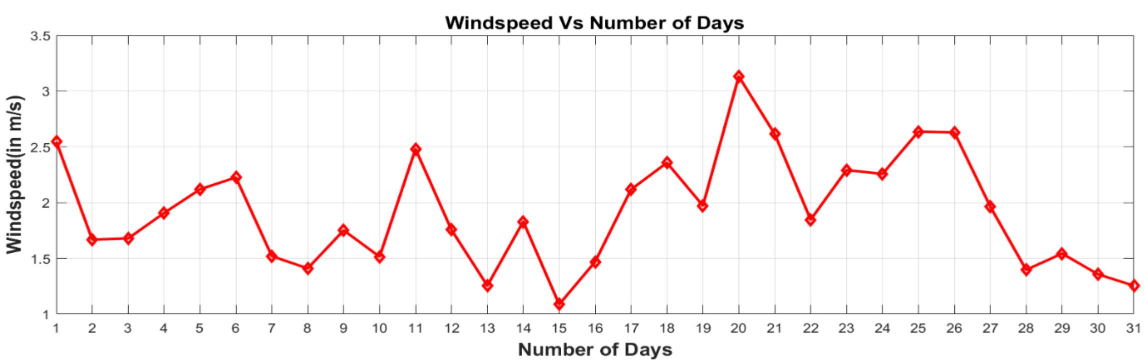



Fig. 4.3. Windspeed vs time for past 31 days

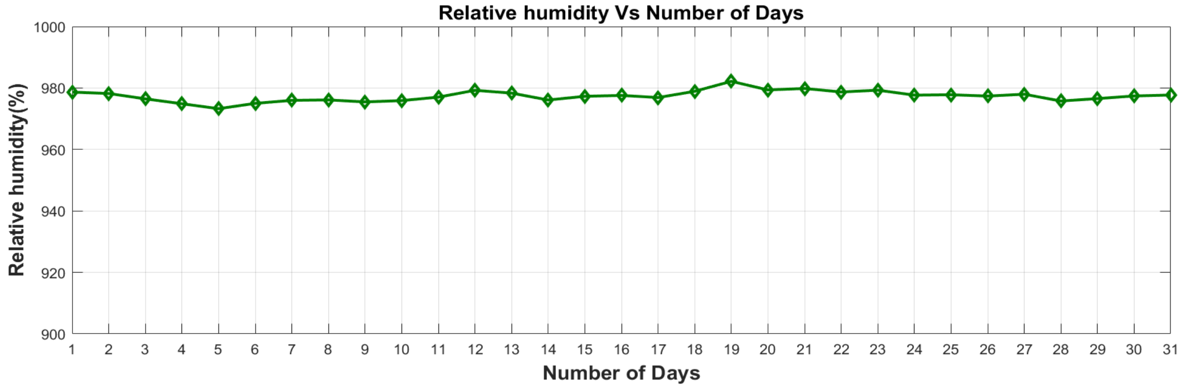



Fig. 4.4. Relative Humidity vs time for past 31 days

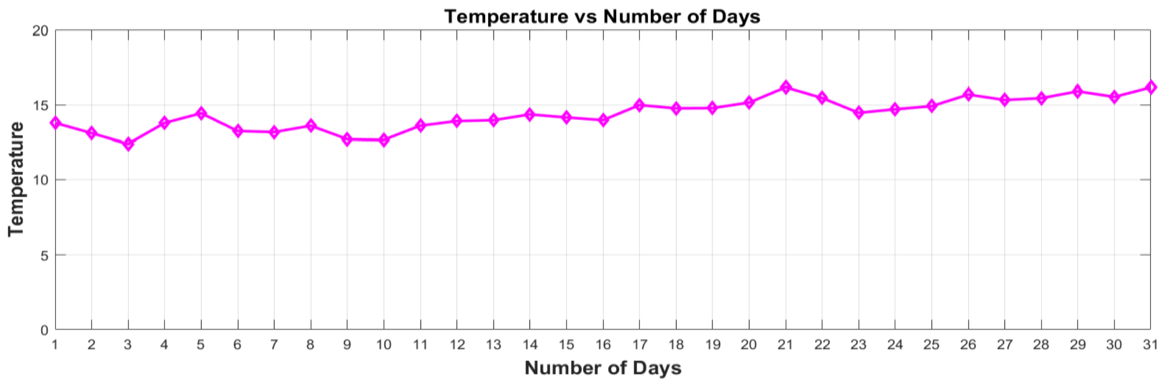



Fig. 4.5. Temperature vs time for past 31 days

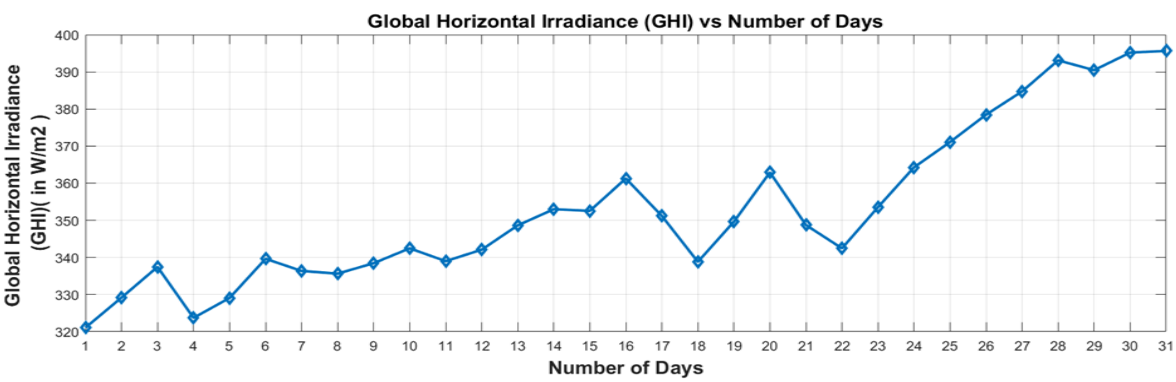
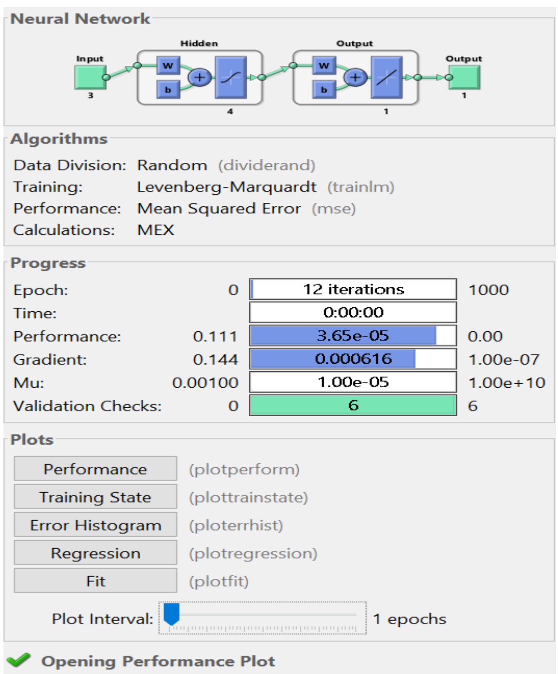



Fig. 4.6. GHI vs time for past 31 days

The nature of windspeed, Relative humidity, temperature and Global horizontal irradiance have been shown in Fig. 4.3-4.6. Here, the previous load, the previous windspeed, the relative humidity, temperature and GHI are the input parameters used for forecasting the load, solar irradiance and windspeed. The nature of the wind is uncertain.

## 4.1 NEXT DAY LOAD FORECASTING USING ANN

A day ahead forecasting has been performed using neural network fitting toolbox. By using the data of Patiala Grid from 1st to 31st January 2017, the load of 1st February is forecasted. The hidden layer has 4 neurons. The ANN model has been trained using LM algorithm as it has a fast convergence speed and requires less time. After the simulation of the model, MSE is obtained for load forecasting is 0.00346573. The ANN model used for the simulation is shown in Fig. 4.7.



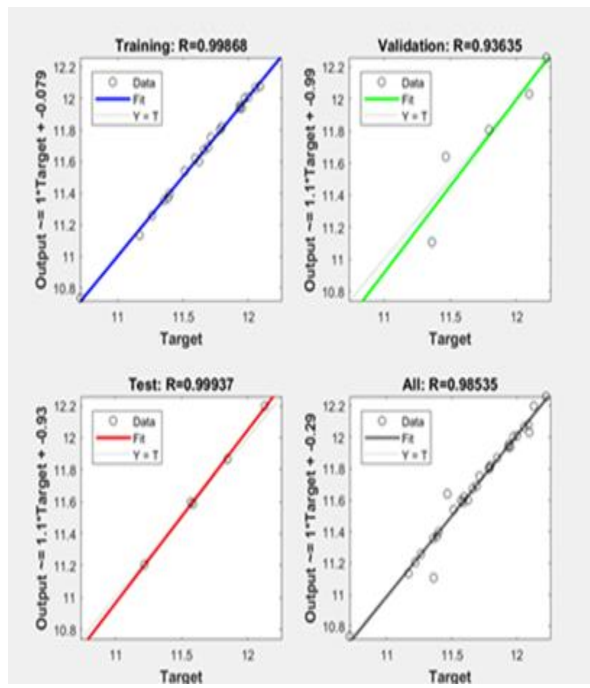




Fig. 4.7 The ANN model

Fig. 4.8. Regression plot

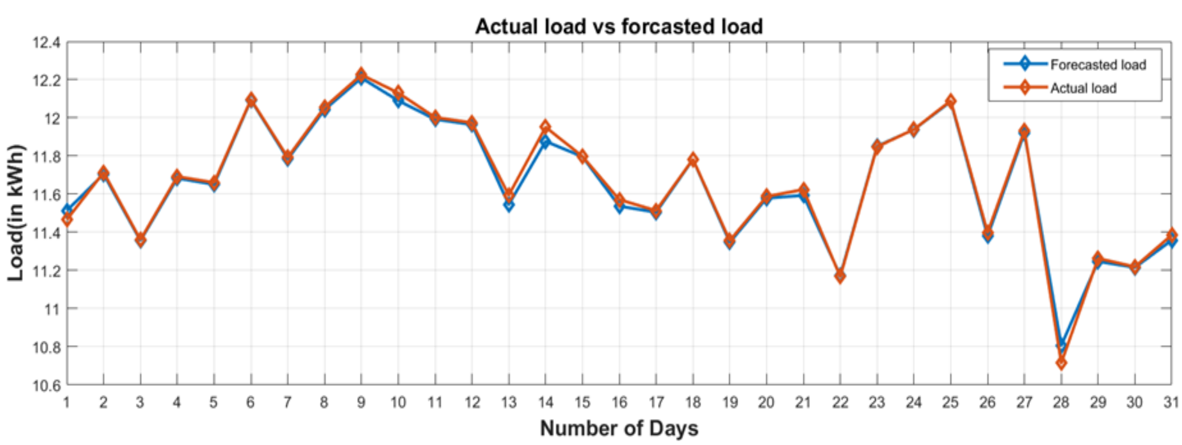



Fig. 4.9. Actual load vs forecasted load based on ANN

From the Fig. 4.8, it can be seen that the regression value for training, validation and test are 0.99868, 0.93635 and 0.99937 respectively. The dashed line represents the target value, the solid line indicates to the best-fit line and black circle represents data. If R value is close to 1 then it indicates that there is close relationship between output and the target,0 means no relationship between them. It can be observed that R value is close to 1 which means that the total response leads to a satisfactory response. The proposed model is simulated using nftool (neural fitting tool). In the Fig. 4.9, the plot of actual load vs forecasted load based on ANN is shown. It can be observed that that the actual load curve is very close to forecasted load which implies that the accuracy of forecasting load using ANN is very high.

## 4.2 NEXT DAY SOLAR IRRADIANCE FORECASTING USING ANN

A day ahead forecasting has been performed using neural network fitting toolbox. By using the data of Patiala Grid obtained from Homer software from 1st to 31st January 2017, the solar irradiance of 1st of Feb is forecasted. The hidden layer has 4 neurons. The ANN model has been trained using LM algorithm as it has a fast convergence speed and requires less time.

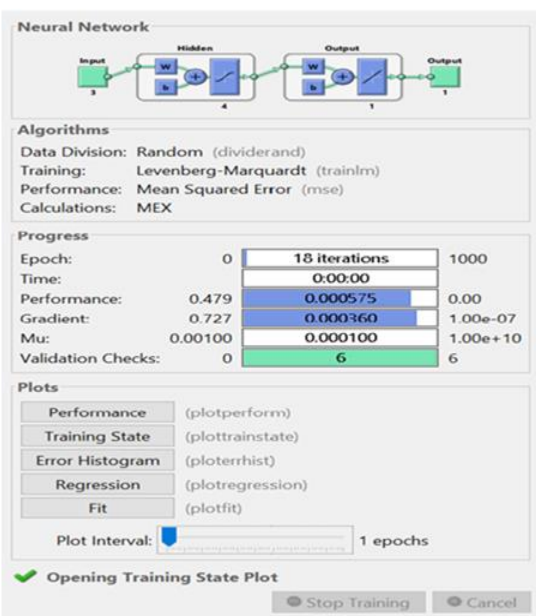



Fig. 4.10. The ANN model for Solar Irradiance Forecasting using ANN

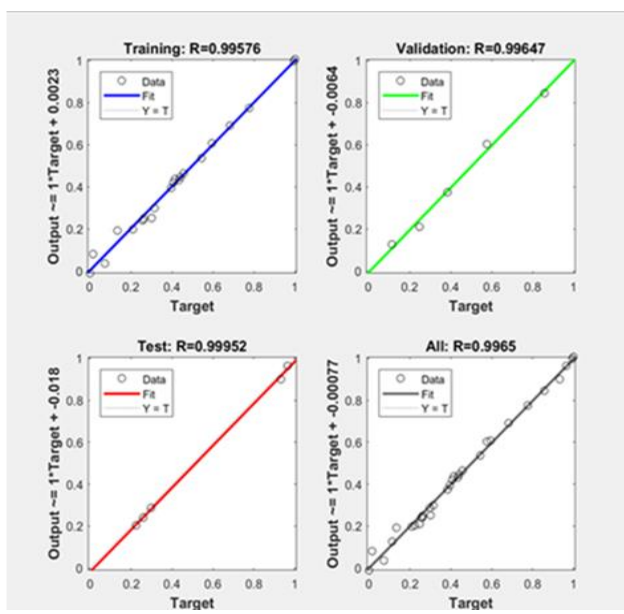



Fig. 4.11. Regression plot for solar irradiance forecasting

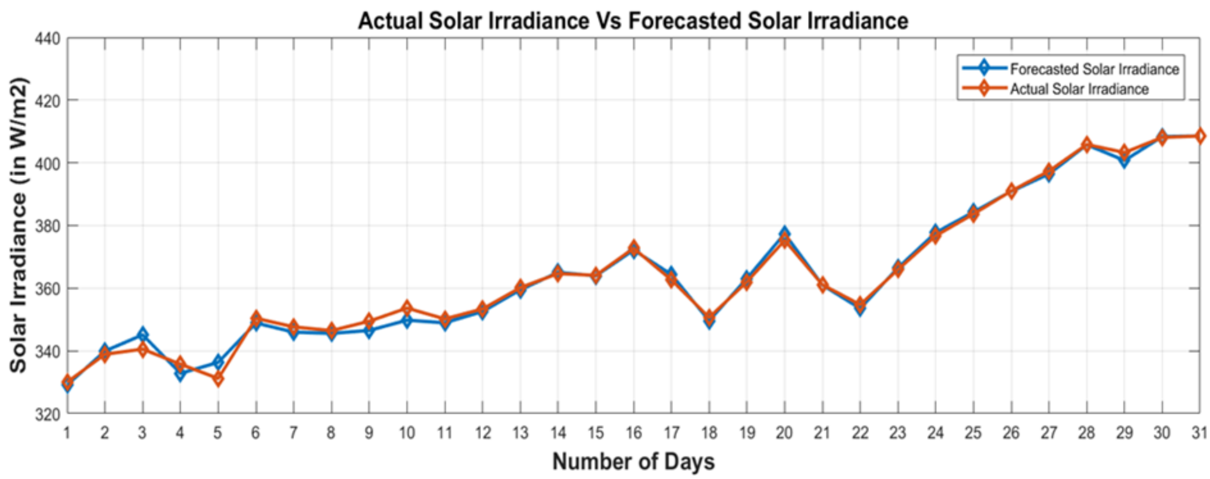



Fig. 4.12. Actual Solar Irradiance vs Forecasted Solar Irradiance based on ANN

After the simulation of the model, MSE is obtained for load forecasting is 0.00064963. The ANN model used for the simulation is shown in Fig. 4.10.

From the Fig. 4.11, it can be seen that the regression value for training, validation and test are 0.99576, 0.99647 and 0.99952 respectively. The dashed line represents the target value, the solid line indicates to the best-fit line and black circle represents data. If R value is close to 1 then it indicates that there is close relationship between output and the target,0 means no relationship between them. In the Fig. 4.12, the plot of actual solar irradiance vs forecasted solar irradiance based on ANN is shown. It can be observed that that the actual solar irradiance curve is very close to forecasted solar irradiance which implies that the accuracy of forecasting load using ANN is very high.

### 4.3 NEXT DAY WIND SPEED FORECASTING USING ANN

A day ahead forecasting has been performed using neural network fitting toolbox. By using the data of Patiala Grid from Homer software from 1st to 31st January 2017, the wind speed of 1st of Feb is forecasted. The hidden layer has 3 neurons. The ANN model has been trained using LM algorithm as it has a fast convergence speed and requires less time. After the simulation of the model, MSE is obtained for load forecasting is 0.022524. The ANN model used for the simulation is shown in Fig. 4.13.

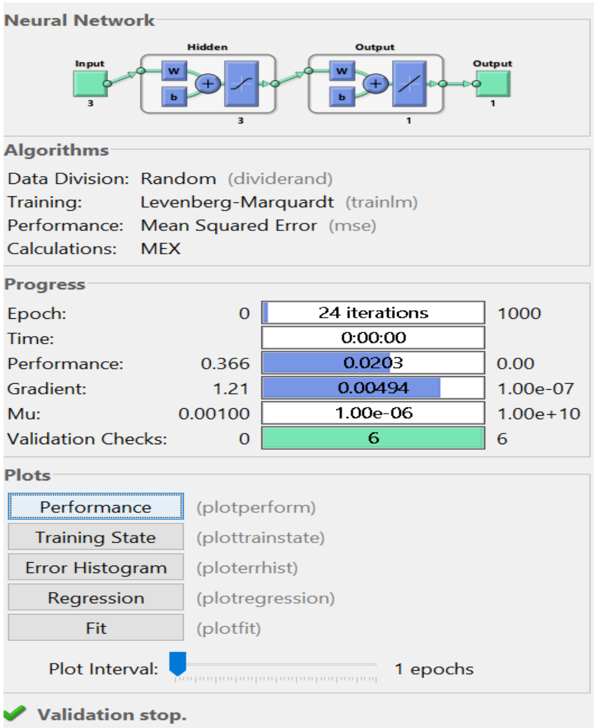



Fig. 4.13. An ANN model for Wind Speed forecasting using ANN

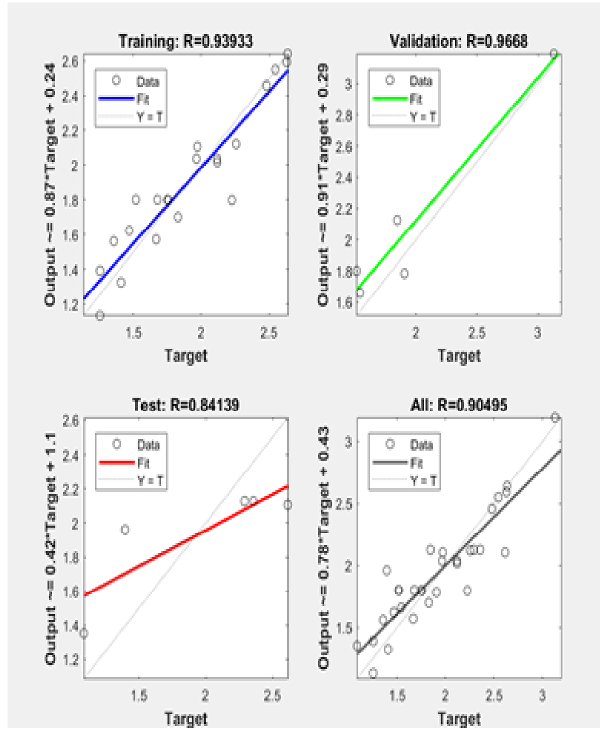



Fig. 4.14. The regression plot for Windspeed forecasting

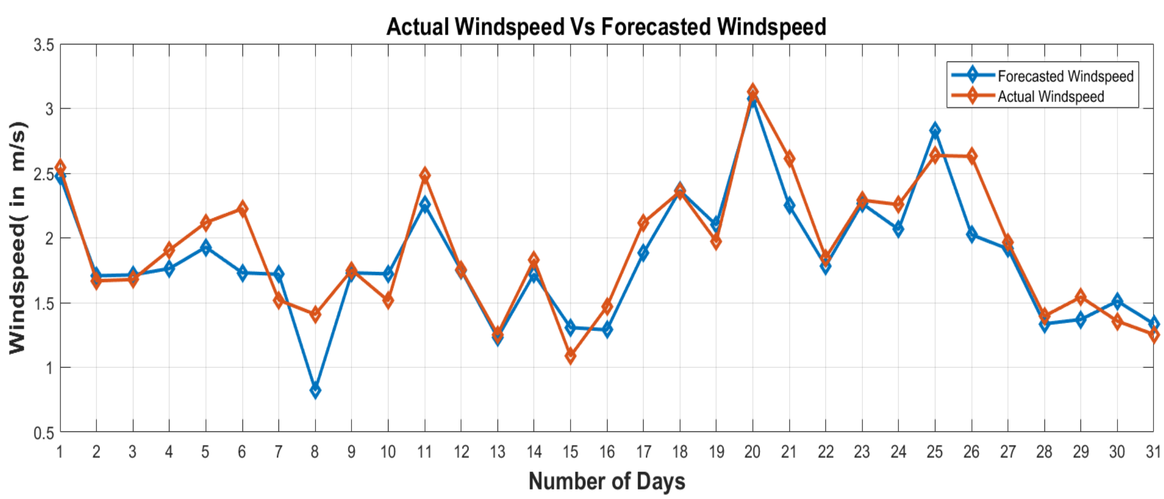



Fig. 4.15. The actual windspeed vs Forecasted Windspeed

From the Fig. 4.14, it can be seen that the regression value for training, validation and test are 0.93933, 0.9668 and 0.84139 respectively. The dashed line represents the target value, the solid line indicates to the best-fit line and black circle represents data. In the Fig. 4.15, the plot of actual wind speed vs forecasted wind speed based on ANN is shown. It can be observed that that the actual wind speed curve is very close to forecasted wind speed which implies that the accuracy of forecasting load using ANN is very high.

## 4.4 FORECASTING USING SLIDING TECHNIQUE

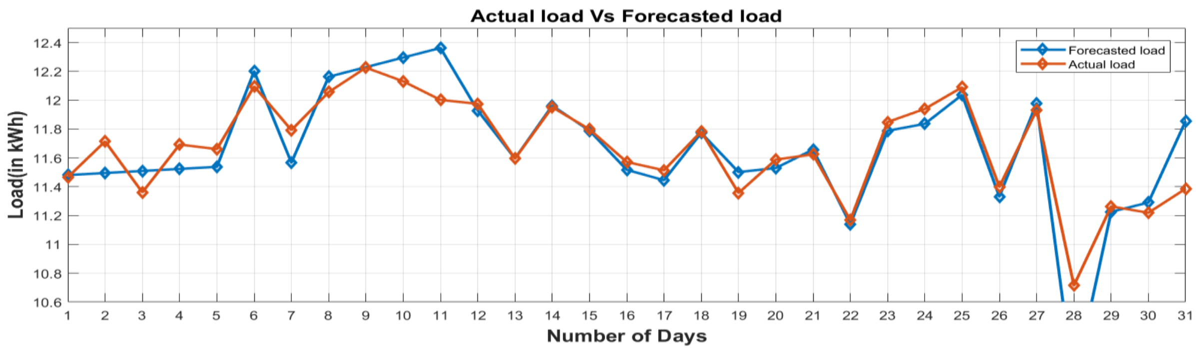



Fig. 4.16. The actual load vs forecasted load based on Sliding Window Approach

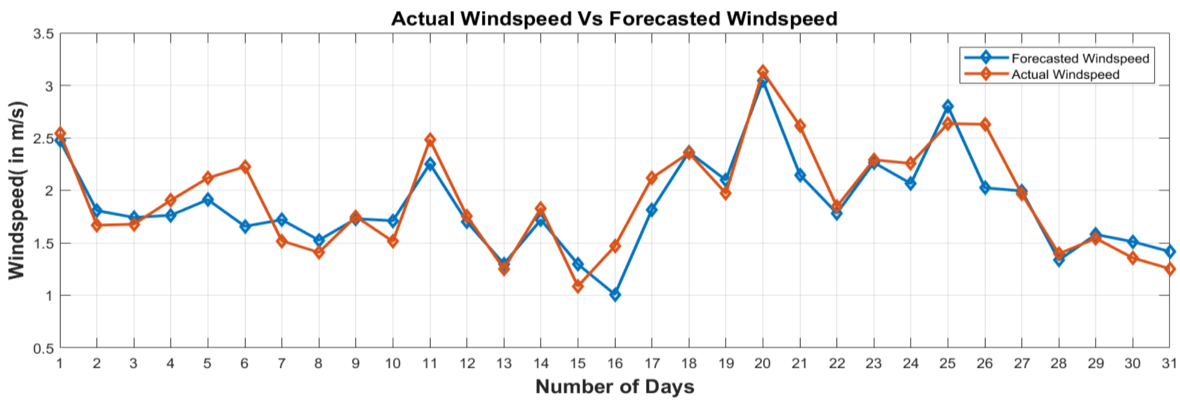



Fig. 4.17. The actual windspeed vs forecasted windspeed based on Sliding Window Approach

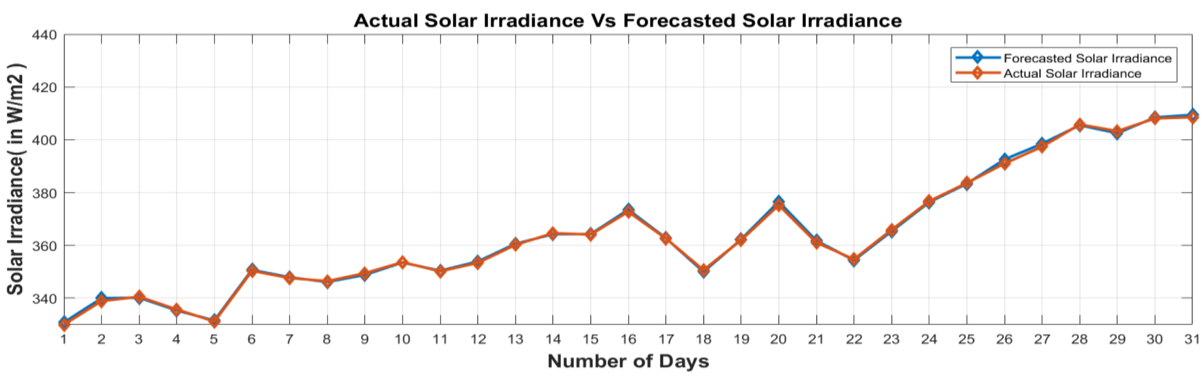



Fig. 4.18. The actual solar irradiance vs forecasted solar irradiance based on Sliding Window Approach

Using Sliding Window Approach, the load has been forecasted for Patiala Grid. In Fig. 4.16, the blue line is a forecasted load using Sliding Window Approach and the orange line is the actual load. Likewise, for Windspeed plot and the solar irradiance plot, the blue line represents the forecasted data obtained from Sliding Window Approach. From the Fig. 4.16, 4.17 and 4.18, it can be seen that since the forecasted line is close to actual data line, the accuracy is high.

### 4.5 COMPARISON OF METHODS FOR FORECASTING

Table 4.1: Comparison of Forecasting Methods

| Methods | MSE (mean squared error) |
|---|---|
| ANN for load forecasting | 0.000347 |
| ANN for Solar irradiance forecasting | 0.000640963 |
| ANN for Windspeed forecasting | 0.0225242 |
| SWA for load forecasting | 0.0356 |
| SWA for Solar Irradiance forecasting | 0.4081 |
| SWA for Windspeed forecasting | 0.2785 |

From the Table 4.1, it can be concluded that the ANN technique is preferable compare to SWA for the given system. The MSE is least in case of ANN compared to SWA. The results in both the cases have been compared and the result is found satisfactory.

# CHAPTER 5

# CONCLUSION AND FUTURE SCOPE

## 5.1 CONCLUSION

The Sliding Window Approach has been studied. The Sliding Window Technique is used for forecasting the data of 1st of February. It is simulated for forecasting the load, solar and wind data of Patiala Grid. Using this same technique,1 month of February data has been forecasted. It implies that the accuracy of forecasting is high. Besides, Artificial neural network is used to forecast 1 month of February using Levenberg Marquart method. For these two methods, the results in both the cases have been compared and the result is found satisfactory. And, it has been observed that the performance and the accuracy is better in case of Artificial Neural Network for the given system.

## 5.2 FUTURE SCOPE

1. To incorporate the Sliding Window algorithm with the Artificial Neural network (ANN) so as to enhance the forecasting.
2. To develop an ANN controller based on Sliding Window approach in microgrid as the concept of sliding window approach is not much implemented in the area of ANN based controller.
3. To focus more on the application of Sliding Window Approach for microgrid management.

## REFERENCES


[1] J.S. Chou and T.K. Nguyen, "Forward forecast of stock price using sliding-window metaheuristic-optimized Machine Learning Regression," *IEEE Transactions on Industrial Informatics*, vol 14, p 3132-3142, 2018.

[2] K.J. Lang and A.H. Waibel, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol 3,,p 23-43,1990.

[3] A.G. Stan, G. Adam and G. Livint, " Multistep-ahead prediction of power demand using a sliding window technique and neural networks," *International Conference and Exposition on Electrical and Power Engineering*, p 54 – 58,2012.

[4] A. Chai, "A time-varying data classification approach based on Ann with sliding window," *Applied Mechanics and Materials*, p 731-735,2013.

[5] M. Akpinar, M.F. Adak and N. Yumusak, "Article day-ahead natural gas demand forecasting using optimized ABC-based neural network with sliding window technique: The Case Study of Regional Basis in Turkey," *Energies*, vol 10, p 1-20,2017.

[6] F. Alfaris, A. Alzahrani and J.W. Kimball, "Stochastic model for PV sensor array data," *International Conference on Renewable Energy Research and Application*, p 798 – 803,2014.

[7] A. Alzahrani, A.Kimball and J.W. Dagli, "Predicting solar irradiance using time series neural networks," *Procedia Computer Science*, vol36, p 623 – 628, 2014.

[8] J. Foster, X. Liu and S.M. Loone, "Adaptive sliding window load forecasting," *Irish Signals and Systems Conference*, ISSC, p 1-6, 2017.

[9] M. Vafaeipour, O Rahbari , M.A Rosen, F. Fazelpour and P. Ansarirad, "Application of sliding window technique for prediction of wind velocity time series," *International Journal of Energy and Environmental Engineering*, vol 5, p 105-108, 2014.

[10] F.M. Diasa, A. Antunesa, J. Vieirab and A. Motac, "A sliding window solution for the on-line implementation of the Levenberg–Marquardt algorithm," Engineering Applications of Artificial Intelligence,vol 19, p 1–7,2006.

[11] I. Gallo, E. Binaghi ,M. Carullo and N. Lamberti . Named entity recognition by neural sliding window. The Eighth IAPR International Workshop on Document Analysis Systems, 87 – 92, 2017.

[12] H.S. Hota, R. Handa and A.K. Shrivas, “Time series data prediction using sliding window based RBF neural network,” *International Journal of Computational Intelligence Research*, vol 13, p 1145-1156, 2017.

[13] R.J. Frank, N. Davey and S.P. Hunt, “Time series prediction and neural networks,” *Journal of Intelligent and Robotic Systems*,vol.31,p 91-103,2001.

[14] K.Andrea , C. Dunis and S. Khalil, “ Modelling, forecasting and trading with a new sliding window approach: the crack spread example,” *International Journal of Computational Intelligence Research*, vol 16, p 1875–1886, 2016.

[15] L.Mozaffari, A.Mozaffari and N.L. Azad, “Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: a case study on San Francisco urban roads. Engineering Science and Technology, *“an International Journal*, vol 11, p 1-13, 2014.

[16] K.Piyush and S.S. Bedi, “ Weather forecasting using sliding window algorithm*,” ISRN Signal Processing*, vol 2013,p 1-5, 2013.

[17] M.Xiangwei, Z. Ping and L.Chunjin, “Sliding Window Algorithm for aircraft landing problem, “ p874 – 879, 2011.

[18] J.Kumar , S.Mishra ,A. Hansdah and R. Kumar, “Modified sliding window algorithm for weather forecasting,” *IEEE International Conference on Technological Innovations in ICT For Agriculture and Rural Development ( TIAR 2016)*,p 175 – 180, 2016.

[19] T. Vigneysha and N. Kumarappan, “Autonomous operation and control of photovoltaic/ solid oxide fuel cell/battery energy storage based microgrid using fuzzy logic controller*,” International Journal of Advanced Research in Computer Science and Software Engineering*, p5-1, 2015.

[20] K. Singh, M.Ugender and DLH Bhavani, “Performance of Micro-grid connected Hybrid Photovoltaic/Fuel cell with ANN control to improve the Power quality,” *Advanced Science and Technology*, p 58,p 75-82 ,2014.

[21] P.C. Sekhar, S. Mishra and R.Sharma, “Data analytics based neuro-fuzzy controller for diesel-photovoltaic hybrid AC microgrid,” *IET Gener. Transm. Distrib*.. vol 9, p193–207,2014.

[22] C.J Huang, “Neural Network Based Microgrid Voltage Control,” *Advanced Science and Technology*, vol.5, p 15-22 ,2013.

[23] H.Bevrani, “An Intelligent Droop Control for Simultaneous Voltage and Frequency Regulation in Islanded Microgrids,” *IEEE transaction on smart grid*, vol 4, p 534-539 ,2013.
[24] S.M.T. Bathaee and M.H. Abdollahi, “Fuzzy-Neural Controller Design for Stability Enhancement of Microgrids,” *IEEE transaction on smart grid*, vol 1, p 134-139, 2007.

# APPENDIX

**SLIDING WINDOW PROGRAM:**

**Forecasting using Sliding Window Approach:**

**Program:**

```
clc;
clear all;
% Current year data
[num,txt,raw] = xlsread('SlidingWindowLoad2017.xlsx');
A=num(1:7,1:3); %Current year data
A1=num(2:8,1:3);
A2=num(3:9,1:3);
A3=num(4:10,1:3);
A4=num(5:11,1:3);
A5=num(6:12,1:3);
A6=num(7:13,1:3);
[num,txt,raw] = xlsread('SlidingWindowLoad2016.xlsx');
B=num(1:14,1:3); %Past year data
B1=num(2:15,1:3);
B2=num(3:16,1:3);
B3=num(4:17,1:3);
B4=num(5:18,1:3);
B5=num(6:19,1:3);
B6=num(7:20,1:3);
%%Sliding Window
SW=zeros(56,3);
W1=zeros(7,3);
W1=num(1:7,1:3);
W2=zeros(7,3);
W2=num(2:8,1:3);
W3=zeros(7,3);
W3=num(3:9,1:3);
W4=zeros(7,3);
W4=num(4:10,1:3);
W5=zeros(7,3);
W5=num(5:11,1:3);
W6=zeros(7,3);
W6=num(6:12,1:3);
W7=zeros(7,3);
```

```
W7=num(7:13,1:3);
W8=zeros(7,3);
W8=num(8:14,1:3);
WW1=[W1;W2;W3;W4;W5;W6;W7;W8];
SW1=zeros(56,3);
W11=zeros(7,3);
W11=num(1:7,1:3);
W21=zeros(7,3);
W21=num(2:8,1:3);
W31=zeros(7,3);
W31=num(3:9,1:3);
W41=zeros(7,3);
W41=num(4:10,1:3);
W51=zeros(7,3);
W51=num(5:11,1:3);
W61=zeros(7,3);
W61=num(6:12,1:3);
W71=zeros(7,3);
W71=num(7:13,1:3);
W81=zeros(7,3);
W81=num(8:14,1:3);
WW2=[W11;W21;W31;W41;W51;W61;W71];
SW2=zeros(56,3);
W12=zeros(7,3);
W12=num(1:7,1:3);
W22=zeros(7,3);
W22=num(2:8,1:3);
W32=zeros(7,3);
W32=num(3:9,1:3);
W42=zeros(7,3);
W42=num(4:10,1:3);
W52=zeros(7,3);
W52=num(5:11,1:3);
W62=zeros(7,3);
W62=num(6:12,1:3);
W72=zeros(7,3);
W72=num(7:13,1:3);
W82=zeros(7,3);
```

```
W82=num(8:14,1:3);
WW3=[W12;W22;W32;W42;W52;W62;W72]
SW2=zeros(56,3);
W13=zeros(7,3);
W13=num(1:7,1:3);
W23=zeros(7,3);
W23=num(2:8,1:3);
W33=zeros(7,3);
W33=num(3:9,1:3);
W43=zeros(7,3);
W43=num(4:10,1:3);
W53=zeros(7,3);
W53=num(5:11,1:3);
W63=zeros(7,3);
W63=num(6:12,1:3);
W73=zeros(7,3);
W73=num(7:13,1:3);
W83=zeros(7,3);
W83=num(8:14,1:3);
WW4=[W13;W23;W33;W43;W53;W63;W73]
SW3=zeros(56,3);
W14=zeros(7,3);
W14=num(1:7,1:3);
W24=zeros(7,3);
W24=num(2:8,1:3);
W34=zeros(7,3);
W34=num(3:9,1:3);
W44=zeros(7,3);
W44=num(4:10,1:3);
W54=zeros(7,3);
W54=num(5:11,1:3);
W64=zeros(7,3);
W64=num(6:12,1:3);
W74=zeros(7,3);
W74=num(7:13,1:3);
W84=zeros(7,3);
W84=num(8:14,1:3);
WW5=[W14;W24;W34;W44;W54;W64;W74]
```

```
SW4=zeros(56,3);
W15=zeros(7,3);
W15=num(1:7,1:3);
W25=zeros(7,3);
W25=num(2:8,1:3);
W35=zeros(7,3);
W35=num(3:9,1:3);
W45=zeros(7,3);
W45=num(4:10,1:3);
W55=zeros(7,3);
W55=num(5:11,1:3);
W65=zeros(7,3);
W65=num(6:12,1:3);
W75=zeros(7,3);
W75=num(7:13,1:3);
W85=zeros(7,3);
W85=num(8:14,1:3);
WW6=[W15;W25;W35;W45;W55;W65;W75]
SW5=zeros(56,3);
W16=zeros(7,3);
W16=num(1:7,1:3);
W26=zeros(7,3);
W26=num(2:8,1:3);
W36=zeros(7,3);
W36=num(3:9,1:3);
W46=zeros(7,3);
W46=num(4:10,1:3);
W56=zeros(7,3);
W56=num(5:11,1:3);
W66=zeros(7,3);
W66=num(6:12,1:3);
W76=zeros(7,3);
W76=num(7:13,1:3);
W86=zeros(7,3);
W86=num(8:14,1:3);
WW7=[W16;W26;W36;W46;W56;W66;W76]
%%%Calculation for the Euclidean distance
val= input('The number of element to be predicted')
```

```
    switch val
    case 1
Sum=0;
ed=zeros(8,1);
ED=zeros(8,1);
sw=0;
for I=1:8
    for j=1:7
for n=1:3
f=((A(j,n)-B(j+sw,n)).^2);
Sum=Sum+sum(f);
end
    end
Sum
ED(I,1)=sqrt(Sum);
ed(I,1)=ed(I,1)+ED(I,1);
sw=sw+1; %Sliding by 1 parameter
end
%%%The corresponding matrix for minimum euclidean distance
[r]=find(ed==min(min(ed)))
        switch [r]
            case r==1
                R=WW1(1:7,:);
             case r==2
                 R=WW1(8:14,:);
            case r==3
                R=WW1(15:21,:);
            case r==4
                R=WW1(22:28,:);
            case r==5
                R=WW1(29:35,:);
            case r==6
                R=WW1(36:42,:);
            case r==7
                R=WW1(43:49,:);
            case r==8
                R=WW1(50:56,:);
        end
```

```
%Computation of the predicted load
VC1=zeros(7,1);
VC2=zeros(6,1);
VP1=zeros(7,1);
VP2=zeros(6,1);
Previous_day_data=A(7,:);
z=0;
z1=zeros(7,1);
for k=1:7
VC1=sum(A,2)./7;
VP1=sum(R,2)./7;
for i=1:6
   VC2(i,1)=VC1(i,1)-VC1(i+1,1);
VP2(i,1)=VP1(i,1)-VP1(i+1,1);
end
Mean1=mean(VC2)
Mean2=mean(VP2)
Predicted_variation=(Mean1+Mean2)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z1(k,1)=Previous_day_data(1,1);
end
z1
   case 2
      Sum1=0;
ed1=zeros(8,1);
ED1=zeros(8,1);
sw=0;
for I=1:8
   for j=1:7
for n=1:3
f1=((A1(j,n)-B1(j+sw,n)).^2);
Sum1=Sum1+sum(f1);
end
   end
Sum1
ED1(I,1)=sqrt(Sum1);
ed1(I,1)=ed1(I,1)+ED1(I,1);
sw=sw+1; %Sliding by 1 parameter
```

```
end
%%%The corresponding matrix for minimum euclidean distance
[r1]=find(ed1==min(min(ed1)))
    switch [r1]
       case r1==1
          R1=WW2(1:7,:);
        case r1==2
           R1=WW2(8:14,:);
       case r1==3
          R1=WW2(15:21,:);
       case r1==4
          R1=WW2(22:28,:);
       case r1==5
          R1=WW2(29:35,:);
       case r1==6
          R1=WW2(36:42,:);
       case r1==7
          R1=WW2(43:49,:);
       case r1==8
          R1=WW2(50:56,:);
    end
%Computation of the predicted load
VC11=zeros(7,1);
VC21=zeros(6,1);
VP11=zeros(7,1);
VP21=zeros(6,1);
Previous_day_data=A1(7,:);
z=0;
z2=zeros(7,1);
for k=1:7
VC11=sum(A1,2)./7;
VP11=sum(R1,2)./7;
for i=1:6
  VC21(i,1)=VC11(i,1)-VC11(i+1,1);
VP21(i,1)=VP11(i,1)-VP11(i+1,1);
end
Mean11=mean(VC21);
Mean21=mean(VP21)
```

```
Predicted_variation=(Mean11+Mean21)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z2(k,1)=Previous_day_data(1,1);
end
z2
   case 3
       Sum2=0;
ed2=zeros(8,1);
ED2=zeros(8,1);
sw=0;
for I=1:8
   for j=1:7
for n=1:3
f2=((A2(j,n)-B2(j+sw,n)).^2);
Sum2=Sum2+sum(f2);
end
   end
Sum2
ED(I,1)=sqrt(Sum2);
ed2(I,1)=ed2(I,1)+ED2(I,1);
sw=sw+1; %Sliding by 1 parameter
end
%%%The corresponding matrix for minimum euclidean distance
[r2]=find(ed2==min(min(ed2)))
       switch [r2]
           case r2==1
               R2=WW3(1:7,:);
            case r2==2
                R2=WW3(8:14,:);
           case r2==3
               R2=WW3(15:21,:);
           case r2==4
               R2=WW3(22:28,:);
           case r2==5
               R2=WW3(29:35,:);
           case r2==6
               R2=WW3(36:42,:);
           case r2==7
```

```
            R2=WW3(43:49,:);
          case r2==8
            R2=WW3(50:56,:);
        end
%Computation of the predicted load
VC12=zeros(7,1);
VC22=zeros(6,1);
VP12=zeros(7,1);
VP22=zeros(6,1);
Previous_day_data=A2(7,:);
z=0;
z3=zeros(7,1);
for k=1:7
VC12=sum(A2,2)./7;
VP12=sum(R2,2)./7;
for i=1:6
    VC22(i,1)=VC12(i,1)-VC12(i+1,1);
VP22(i,1)=VP12(i,1)-VP12(i+1,1);
end
Mean12=mean(VC22)
Mean22=mean(VP22)
Predicted_variation=(Mean12+Mean22)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z3(k,1)=Previous_day_data(1,1);
end
z3
    case 4
        Sum3=0;
ed3=zeros(8,1);
ED3=zeros(8,1);
sw=0;
for I=1:8
    for j=1:7
for n=1:3
f3=((A3(j,n)-B3(j+sw,n)).^2);
Sum3=Sum3+sum(f3);
end
    end
```

```
Sum3
ED3(I,1)=sqrt(Sum3);
ed3(I,1)=ed3(I,1)+ED3(I,1);
sw=sw+1; %Sliding by 1 parameter
end
%%%The corresponding matrix for minimum euclidean distance
[r3]=find(ed3==min(min(ed3)))
    switch [r3]
      case r3==1
        R3=WW4(1:7,:);
       case r3==2
         R3=WW4(8:14,:);
      case r3==3
        R3=WW4(15:21,:);
      case r3==4
        R3=WW4(22:28,:);
      case r3==5
        R3=WW4(29:35,:);
      case r3==6
        R3=WW4(36:42,:);
      case r3==7
        R3=WW4(43:49,:);
      case r3==8
        R3=WW4(50:56,:);
    end
%Computation of the predicted load
VC13=zeros(7,1);
VC23=zeros(6,1);
VP13=zeros(7,1);
VP23=zeros(6,1);
Previous_day_data=A3(7,:);
z=0;
z4=zeros(7,1);
for k=1:7
VC13=sum(A3,2)./7;
VP13=sum(R3,2)./7;
for i=1:6
  VC23(i,1)=VC13(i,1)-VC13(i+1,1);
```

```
VP23(i,1)=VP13(i,1)-VP13(i+1,1);
end
Mean13=mean(VC23);
Mean23=mean(VP23);
Predicted_variation=(Mean13+Mean23)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z4(k,1)=Previous_day_data(1,1);
end
z4
    case 5
        Sum4=0;
ed4=zeros(8,1);
ED4=zeros(8,1);
sw=0;
for I=1:8
    for j=1:7
for n=1:3
f4=((A4(j,n)-B4(j+sw,n)).^2);
Sum=Sum+sum(f4);
end
    end
Sum4
ED4(I,1)=sqrt(Sum4);
ed4(I,1)=ed4(I,1)+ED4(I,1);
sw=sw+1; %Sliding by 1 parameter
end
%%%The corresponding matrix for minimum euclidean distance
[r4]=find(ed4==min(min(ed4)))
        switch [r4]
            case r4==1
                R4=WW5(1:7,:);
             case r4==2
                 R4=WW5(8:14,:);
            case r4==3
                R4=WW5(15:21,:);
            case r4==4
                R4=WW5(22:28,:);
            case r4==5
```

```
            R4=WW5(29:35,:);
          case r4==6
            R4=WW5(36:42,:);
          case r4==7
            R4=WW5(43:49,:);
          case r4==8
            R4=WW5(50:56,:);
        end
%Computation of the predicted load
VC14=zeros(7,1);
VC24=zeros(6,1);
VP14=zeros(7,1);
VP24=zeros(6,1);
Previous_day_data=A4(7,:);
z=0;
z5=zeros(7,1);
for k=1:7
VC14=sum(A4,2)./7;
VP14=sum(R4,2)./7;
for i=1:6
  VC24(i,1)=VC14(i,1)-VC14(i+1,1);
VP24(i,1)=VP14(i,1)-VP14(i+1,1);
end
Mean14=mean(VC24);
Mean24=mean(VP24);
Predicted_variation=(Mean14+Mean24)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z5(k,1)=Previous_day_data(1,1);
end
z5
  case 6
    Sum5=0;
ed5=zeros(8,1);
ED5=zeros(8,1);
sw=0;
for I=1:8
  for j=1:7
for n=1:3
```

```
f5=((A5(j,n)-B5(j+sw,n)).^2);
Sum5=Sum5+sum(f5);
end
  end
Sum5
ED5(I,1)=sqrt(Sum5);
ed5(I,1)=ed5(I,1)+ED5(I,1);
sw=sw+1; %Sliding by 1 parameter
end
%%%The corresponding matrix for minimum euclidean distance
[r5]=find(ed==min(min(ed5)))
    switch [r5]
      case r5==1
        R5=WW6(1:7,:);
       case r5==2
         R5=WW6(8:14,:);
      case r5==3
        R5=WW6(15:21,:);
      case r5==4
        R5=WW6(22:28,:);
      case r5==5
        R5=WW6(29:35,:);
      case r5==6
        R5=WW6(36:42,:);
      case r5==7
        R5=WW6(43:49,:);
      case r5==8
        R5=WW6(50:56,:);
    end
%Computation of the predicted load
VC15=zeros(7,1);
VC25=zeros(6,1);
VP15=zeros(7,1);
VP25=zeros(6,1);
Previous_day_data=A5(7,:);
z=0;
z6=zeros(7,1);
for k=1:7
```

```
VC15=sum(A5,2)./7;
VP15=sum(R5,2)./7;
for i=1:6
   VC25(i,1)=VC15(i,1)-VC15(i+1,1);
VP25(i,1)=VP15(i,1)-VP15(i+1,1);
end
Mean15=mean(VC25);
Mean25=mean(VP25);
Predicted_variation=(Mean15+Mean25)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z6(k,1)=Previous_day_data(1,1);
end
z6
   case 7
      Sum6=0;
ed6=zeros(8,1);
ED6=zeros(8,1);
sw=0;
for I=1:8
   for j=1:7
for n=1:3
f6=((A6(j,n)-B6(j+sw,n)).^2);
Sum6=Sum6+sum(f6);
end
   end
Sum6
ED6(I,1)=sqrt(Sum6);
ed6(I,1)=ed6(I,1)+ED6(I,1);
sw=sw+1; %Sliding by 1 parameter
end
%%%The corresponding matrix for minimum euclidean distance
[r6]=find(ed6==min(min(ed6)))
      switch [r6]
         case r6==1
            R6=WW7(1:7,:);
          case r6==2
             R6=WW7(8:14,:);
         case r6==3
```

```
            R6=WW7(15:21,:);
          case r6==4
            R6=WW7(22:28,:);
          case r6==5
            R6=WW7(29:35,:);
          case r6==6
            R6=WW7(36:42,:);
          case r6==7
            R6=WW7(43:49,:);
          case r6==8
            R6=WW7(50:56,:);
      end
%Computation of the predicted load
VC16=zeros(7,1);
VC26=zeros(6,1);
VP16=zeros(7,1);
VP26=zeros(6,1);
Previous_day_data=A6(7,:);
z=0;
z7=zeros(7,1);
for k=1:7
VC16=sum(A6,2)./7;
VP16=sum(R6,2)./7;
for i=1:6
  VC26(i,1)=VC16(i,1)-VC16(i+1,1);
VP26(i,1)=VP16(i,1)-VP16(i+1,1);
end
Mean16=mean(VC26);
Mean26=mean(VP26);
Predicted_variation=(Mean16+Mean26)*0.5;
Previous_day_data=Previous_day_data+Predicted_variation;
z7(k,1)=Previous_day_data(1,1);
end
z7
end
Time=1:1:7;
figure(1);
[num,txt,raw] = xlsread('SlidingWindowLoad2017.xlsx');
```

```
Actual_data=num(1:7,1)
switch val
   case 1
       plot(Time,z1);
title('Predicted load vs time');
xlabel('Time');
ylabel('Load')
   case 2
       plot(Time,z2);
title('Predicted load vs time');
xlabel('Time');
ylabel('Load')
   case 3
       plot(Time,z3);
title('Predicted load vs time');
xlabel('Time');
ylabel('Load')
   case 4
       plot(Time,z4);
title('Predicted load vs time');
xlabel('Time');
ylabel('Load')
   case 5
       plot(Time,z5);
title(' Predicted load vs time');
xlabel('Time');
ylabel('Load')
   case 6
       plot(Time,z6);
title(' Predicted load vs time');
xlabel('Time');
ylabel('Load')
   case 7
plot(Time,z7);
title('Predicted load vs time');
xlabel('Time');
ylabel('Load');
end
```

**Result:**

**WW1 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |

| | | |
|---|---|---|
| **10.2146** | **15.4590** | **975.7540** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |
| **10.8858** | **16.1830** | **977.7140** |

**WW2 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |

| | | |
|---|---|---|
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |

**WW3 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |

| | | |
|---|---|---|
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |

**WW4 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |

| | | |
|---|---|---|
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |

**WW5 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |

| | | |
|---|---|---|
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |

| | | |
|---|---|---|
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |

**WW6 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |

| | | |
|---|---|---|
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |

**WW7 =**

| | | |
|---|---|---|
| **11.2833** | **14.7710** | **978.9180** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **10.8563** | **14.7900** | **982.1640** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |

| | | |
|---|---|---|
| **11.5896** | **14.9210** | **977.8020** |
| **11.0879** | **15.1620** | **979.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.1242** | **16.1910** | **979.8410** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.6675** | **15.4700** | **978.6960** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **11.3483** | **14.4700** | **979.2850** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **11.4396** | **14.7100** | **977.6760** |
| **11.5896** | **14.9210** | **977.8020** |
| **10.8988** | **15.7020** | **977.3700** |
| **11.4321** | **15.3380** | **977.9690** |
| **10.2146** | **15.4590** | **975.7540** |
| **10.7633** | **15.9200** | **976.5510** |
| **10.7192** | **15.5470** | **977.4030** |

**The number of element to be predicted1**

```
val =

     1

z1 =

   10.8939
   10.9019
   10.9100
   10.9180
   10.9261
   10.9341
   10.9422
>>
The number of element to be predicted2

val =

     2

r1 =

     1

z2 =

   11.4806
   11.4949
   11.5091
   11.5234
   11.5377
   11.5520
   11.5662
```

For forecasting the wind speed and solar irradiance, same programming is applied with only the changes in input data.

# PLAGIARISM REPORT

Report

ORIGINALITY REPORT

| 12% | 3% | 12% | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Piyush Kapoor, Sarabjeet Singh Bedi. "Weather Forecasting Using Sliding Window Algorithm", ISRN Signal Processing, 2013<br>Publication | 1% |
| 2 | www.ijer.in<br>Internet Source | 1% |
| 3 | Ali R. Tahavvor, Mahmood Yaghoubi. "Prediction of frost deposition on a horizontal circular cylinder under natural convection using artificial neural networks", International Journal of Refrigeration, 2011<br>Publication | 1% |
| 4 | Shady Mahmoud Elgarhy, Mahmoud M. Othman, Adel Taha, Hany M. Hasanien. "Short term load forecasting using ANN technique", 2017 Nineteenth International Middle East Power Systems Conference (MEPCON), 2017<br>Publication | 1% |
| 5 | Sekhar, P.C., Sukumar Mishra, and Rishi Sharma. "Data analytics based neuro-fuzzy | 1% |