

A Comparative study on Different ANN Techniques in Wind Speed Forecasting for Generation of Electricity

Ranvijay Parmar¹, Manish Shah² MIEEE, MIE, MISLE, Mona Gupta Shah³

¹(M.Tech Scholar, Department of Electrical & Electronics, VITS Indore, India. ranvijayparmar20@gmail.com)

²(Assistant Professor, Department of Electrical & Electronics, VITS Indore, India. ermanish777@gmail.com)

³(Assistant Professor, Department of Physics Govt. Degree College, Bhabar, Uttarakhand, India.
mona_gupta123@rediffmail.com)

Abstract: There are several available renewable sources of energy, among which Wind Power is the one which is most uncertain in nature. This is because wind speed changes continuously with time leading to uncertainty in availability of amount of wind power generated. Hence, a short-term forecasting of wind speed will help in prior estimation of wind power generation availability for the grid and economic load dispatch. This paper presents a comparative study of a Wind speed forecasting model using Artificial Neural Networks (ANN) with three different learning algorithms. ANN is used because it is a non-linear data driven, adaptive and very powerful tool for forecasting purposes. Here an attempt is made to forecast Wind Speed using ANN with Levenberg-Marquard (LM) algorithm, Scaled Conjugate Gradient (SCG) algorithm and Bayesian Regularization (BR) algorithm and their results are compared based on their convergence speed in training period and their performance in testing period on the basis of Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Square Error (MSE). A 48 hour ahead wind speed is forecasted in this work and it is compared with the measured values using all three algorithms and the best out of the three is selected based on minimum error.

Keywords: Wind Speed Prediction, ANN, trainlm, trainscg, trainbr.

I. Introduction

Currently most of the countries in the world are focusing in increasing power generation through renewable sources in order to tackle high carbon emission while committing energy security [1]. According to [2], 76% of global energy is supplied using Non-Renewable Energy Resources (fossil fuels + Nuclear) and rest 24% is generated using Renewable Sources. Out of renewable sources, hydropower is leading with 16.6% out of 24% is generated by it. After Hydro comes the wind and solar power with the contribution of 3.7% and 1.3%.

Out of all renewable sources of energy wind energy is the one which is most uncertain in nature. This is because wind speed changes continuously with time due various natural parameters, leading to uncertainty in availability of amount of wind power that can be generated using it. If we integrate this wind power system directly to the existing power system, it will lead to a number of issues in terms of attaining good power quality, power system stability, frequency of generated power, rated terminal voltage, optimizing spinning reserve capacity, uncertainty in wind power in to unit commitment and reducing power dispatching issues in the grid power balance, and, then we have planning and economic problems including, economic load dispatching [22].

Although in 2015, renewable energy shares only 24% of Global energy consumption due to the fact that renewable energy sources are unpredictable in nature specially Wind (speed) and they are also costly compare to coal diesel because of very costly technology required. Hence in order to increase generating capacity and efficiency of renewable power plants this territory is required to be explored more. The global contribution of Wind energy at the end of 2015 was 432.9 GW, representing cumulative market growth of more than 17 %. India stands 4th in the World in terms of Wind Power Capacity/Generation and 5th in Annual investment in Renewable energy resources [2]. Now focusing on present scenario of Renewable energy in India, present capacity of wind power in India is about 302251 MW out of which only 8.86 % Wind Power potential is utilized. Means Wind Power has large potential to improve India's power generation capacity if more investment is done in it. The only obstacle in this is the uncertainty caused by the discontinuous nature of wind energy which affects the power grid which demotivates investors to invest in this field.

Hence, while harnessing wind Power, it is important to forecasting wind speed for energy managers and electricity traders, to eliminate the risks of unpredictability and to perform efficient electrical load dispatch [3][4]. Although certain methodologies are developed like Statistical Methods, Fuzzy System Based Models, Artificial Intelligence Techniques, Evolutionary Algorithms Based Techniques etc. for forecasting [5] but an accurate forecasting is important since wind power is directly proportional to cube of wind speed, hence any error in wind speed will lead to cube of that error in wind power. The following relation gives the output power of variable-speed wind turbine:

$$P_m = \frac{1}{2} \rho \pi R^2 v^3 C_p \tag{1}$$

Where v stands for wind speed (m/s), R is the radius of the rotor (m), ρ is the air density (kg/m³), and C_p stands for maximum value of rotor efficiency for each steady wind speed [6]. Here in this study we are utilizing Artificial Neural Network (ANN) of Artificial Intelligence Techniques because unlike statistical methods, ANN models are simpler to construct and require shorter development time and these don't require to explicitly defining mathematical expressions [5]. A large set of wind speed data measured at 1 hour intervals is utilized as input in algorithm development. This data is then fed to our proposed ANN as training and testing data. On the basis of this data ANN will forecast wind speed. Forecasted wind speed values can be utilized in evaluating Wind Power potential and Wind Power planning. Performance Functions used in present Neural Network to verify results are following. Mean absolute error (MAE) is used to measure how close predicted speeds are to the actual speeds. The mean absolute error is given by:

$$MAE = \frac{1}{N} \sum_{t=1}^N |A_t - \hat{A}_t| \tag{2}$$

The mean absolute percentage error (MAPE), is also used to measure prediction accuracy of a forecasting method. It expresses accuracy as a percentage, and is given by:

$$MAPE = \frac{100}{N} \sum_{t=1}^N \frac{|A_t - \hat{A}_t|}{A_t} \tag{3}$$

Where, N is the number of fitted points, A_t is the actual value and \hat{A}_t is the forecasted value [7].

The paper has been organized in five sections. Section II presents the Methodology used. Section III discusses the various Data and Model of ANN for Wind Speed Forecasting. Results are presented in Section IV. Section V discusses the Conclusion and Future work.

II. Methodology

Artificial Neural Networks (ANN) are computing systems or technique that are inspired by the learning architecture of human brain to discover the relations between the input and target variables of a system. Human brain consists of a large set of structural constituents, known as neurons, which form a well-connected network to respond to an input signal to perform all its computations / calculations in a certain complex task such as image and voice recognition task and they do this with incredible speed and accuracy. Neurons are simple processing units, which has the ability to store experimental data and which work as parallelly distributed processor. A network of connected artificial neurons can be designed, and a learning algorithm can be applied to train it [8]. Signals (Input data) are passed between neurons over connection links and Each connection link has an associated weight, which in a neural network, multiplies the signal transmitted. The weights represent information being used by the network to solve a problem. Then the weighted sum is operated upon by an activation function (usually nonlinear), and output data are conveyed to other neurons. The weights are continuously altered while training to improve accuracy and generalize abilities [9] [10].

2.1 Levenberg–Marquardt (LM) Algorithm

Reducing error function is the main reason to use this algorithm. Levenberg-Marquardt algorithm is a very efficient technique for minimizing a nonlinear function. The algorithm includes many different variables like in present study we have output data, weight between neurons and error function, that determine efficiency and success rate of model. The ideal values of these variables are very dependent on the test function [11] [12]. Levenberg-Marquardt algorithm is fast and has stable convergence [13]. This algorithm was designed to approach second-order training speed without computing the Hessian matrix. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated and the gradient can be computer as:

$$H = J_k^T J_k \tag{4}$$

$$g = J_k^T e \tag{5}$$

Where J_k is the Jacobian matrix for k^{th} input, which contains first order derivatives of the network errors with respect to the weights and biases, e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix [14].

The Levenberg –Marquardt algorithm is actually a blend of the steepest descent method and the Gauss–Newton algorithm. The following is the relation for LM algorithm computation:

$$W_{k+1} = W_k - [J_k^T J_k + \mu I]^{-1} J_k^T e_k \tag{6}$$

where I is the identity matrix, W_k is the current weight, W_{k+1} is the next weight, e_{k+1} is the current total error, and e_k is the last total error, μ is combination coefficient [14] [15]. It tries to combine the advantages of both the methods hence it inherits the speed of the Gauss–Newton algorithm and the stability of the steepest descent method. The combination coefficient μ is multiplied by some factor (β) whenever a step would result in an increased e_{k+1} and when a step reduces e_{k+1} , μ is divided by β . In this study, we used $\beta=10$. When μ is large the algorithm becomes steepest descent while for small μ the algorithm becomes Gauss-Newton. [14] In present study, the Levenberg-Marquardt (LM) learning algorithm was applied with in the input Neurons have no transfer function. The logistic sigmoid transfer (logsig) and linear transfer (purelin) functions were used in the hidden and output layers of the network as an activation function, respectively.

2.2 Scaled Conjugate Gradient (SCG) Algorithm

Conjugate Gradient (CG) algorithm is a modified version of steepest descent algorithm. In CG, a hunt is done in such a direction so as to generate a faster convergence than the steepest decent direction, while saving the error minimization attained in all previous steps. [16]. this direction is called the conjugate direction. In most of the CG algorithms the step-size is adjusted at each iteration. The algorithm starts in the direction of the steepest descent during first iteration given by the negative of the gradient as in equation (7):

$$p_0 = -g_0 \tag{7}$$

The weights are updated as follows:

$$x_{k+1} = x_k + \alpha_k g_k \tag{8}$$

Where, α_k is the determined step size and the next step size is calculated as:

$$P_k = -g_k + \beta_k P_{k-1} \tag{9}$$

Where p is search direction vector and g is gradient direction vector.

There are various versions of Conjugate Gradient algorithms which can be categorized by the manner in which the factor β_k is calculated [16]. In this study, we have used Scaled Conjugate Gradient (SCG) Algorithm, in which we use LM algorithm combined with CG algorithm to calculate step size, unlike only the line search technique in the CG approach. In the Scaled Conjugate Gradient algorithms, a search is performed along the direction in which the performance function decreases rapidly, while preserving the error minimization. At each iteration, the step size is adjusted and a search is made along the gradient direction to calculate the step size. Generally, the line search algorithm is used to determine the step size. This line search makes the scaled conjugate gradient descent algorithms time consuming as it is required in all iterations [17]. In this model, the Levenberg-Marquardt (LM) learning algorithm was applied. Neurons in the input layer have no transfer function to determine the step size. The logistic sigmoid transfer (logsig) and linear transfer (purelin) functions were used in the hidden and output layers of the network as an activation function, respectively. For SCG, β_k factor calculation and direction of the new search can be shown as in following equations [18]:

$$\beta_k = \frac{(|g_{k+1}|^2 - g_{k+1}^T g_k)}{g_k^T g_k} \tag{10}$$

&

$$P_{k+1} = -g_{k+1} + \beta_k P_k \tag{11}$$

Design parameters are updated at each iteration user independently, which is crucial for the success of the algorithm. This is a major advantage compared to the line search based algorithms.

2.3 Bayesian Regularization (BR) Algorithm

BRANNs are extra vigorous than usual back-propagation networks and can lessen the necessity for long cross-validation. BR algorithm is a process that changes a nonlinear regression into a “well-modelled” statistical problem in the means of a ridge regression. In this algorithm regularization is used to improve the network by optimizing the performance function ($F(\omega)$). The performance function $F(\omega)$ is the sum of the squares of the errors of the network weights (E_w) and the sum of squares error of the data (E_D) [19] [20]:

$$F(\omega) = \alpha E_w + \beta E_D \tag{12}$$

Where

$$E_w = \sum_{i=1}^n w_i^2 \quad \& \quad E_D = \sum_{k=1}^n e_k^2 \tag{13}$$

Both α and β are the objective function parameters. In the BR framework, the weights of the network are viewed as random variables, and then the distribution of the network weights and training set are considered as Gaussian distribution. The α and β factors are defined using the Bayes’ theorem. The Bayes’ theorem relates two variables (or events), A and B, based on their prior (or marginal) probabilities and posterior (or conditional) probabilities as [21]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{14}$$

Where $P(A|B)$ is the conditional probability of event A, depending on event B, $P(B|A)$ the conditional probability of event B, depending on event A, and $P(B)$ the previous probability of event B. In order to get the best values of weights, performance function $F(\omega)$ (12) needs to be minimized, which is the equal to maximizing the following probability function given as:

$$P(\alpha, \beta|D, M) = \frac{P(D|\alpha, \beta, M) P(\alpha, \beta|M)}{P(D|M)} \quad (15)$$

Where α and β are, the factors on which value of performance function is dependent and is the one which is needed be to optimized, M is the particular neural network architecture, D is the weight distribution, $P(D|M)$ is the normalization factor, $P(D|\alpha, \beta, M)$ is the likelihood function of D for given α, β, M and $P(\alpha, \beta|M)$ is the unvarying prior density for the regularization parameters. As a result of this method, finest values for α and β for a given weight space are founded. Later, algorithm travels into LM phase in which Hessian matrix calculations takes place and updates the weights in the direction so as to minimize the value of objective function. Then, if the convergence of objective function does not meet according to expectation, algorithm estimates new values for α and β and the whole procedure repeats itself until the required convergence is achieved [20].

III. Data Inputs

The main ingredient in forecasting using ANN is historical data set. Based on this historical data weights and bias are calculated in training stage. For a proper training of network, accurate value of all the parameters are required otherwise it will lead to wrong training of network which will result in incorrect weight value and finally inefficient forecasting. Hence data should be processed before feeding it to network as an input. Following inputs elements are taken for forecasting wind speed:

- Temperature
- Pressure
- Humidity
- Dew Point
- Season
- Time (hour)
- Last one day average wind speed
- Last one week average speed
- Last week wind speed on same day same time.

3.1 Data Refining

Refining data is very essential before initiating any type of data-analysis study. Refining is the process of detecting and correcting corrupt or inaccurate values from dataset. The discrepancies observed in the given datasets are removed and the resulting refined time-series for each of the weather parameters is then ready for use.

3.2 Normalization of the Data

The input to the network should be the normalized version of recorded values of weather parameters. This is done because different parameters have different scales, which will reduce speed and training network with raw data is going to produce negative impact on training output. After normalization, scales of all the weather parameter will become same. The normalized can either make values of parameters ranging in between -1 to 1 or in between 0 to 1. After training with this normalized values, obtained output is again de-normalized to get the actual forecast value. We have used min-max normalization method in present study. This method normalizes the values of any parameter according to their minimum and maximum values. All the parameters except temperature were normalized to the [0,1] range, while temperature was in the range of [-1,1], given by:

$$N = low + \frac{(high-low)(a-min)}{max-min} \quad (16)$$

Where N is normalized value of parameters and a is actual value.

The Neural Network in present study consists of three layers. The first one is Input layer (consist of 9 Neurons for 9 input element) from where inputs are feed to model for further computation. Then comes second layer called Hidden Layer (consist of 15 Neurons), this is where activation function is used to limit value of output Neuron. At last we have output layer (1 Neuron) form which we take output result for comparison with actual result to calculate error and the feeding it back to model to vary weight accordingly for improving performance.

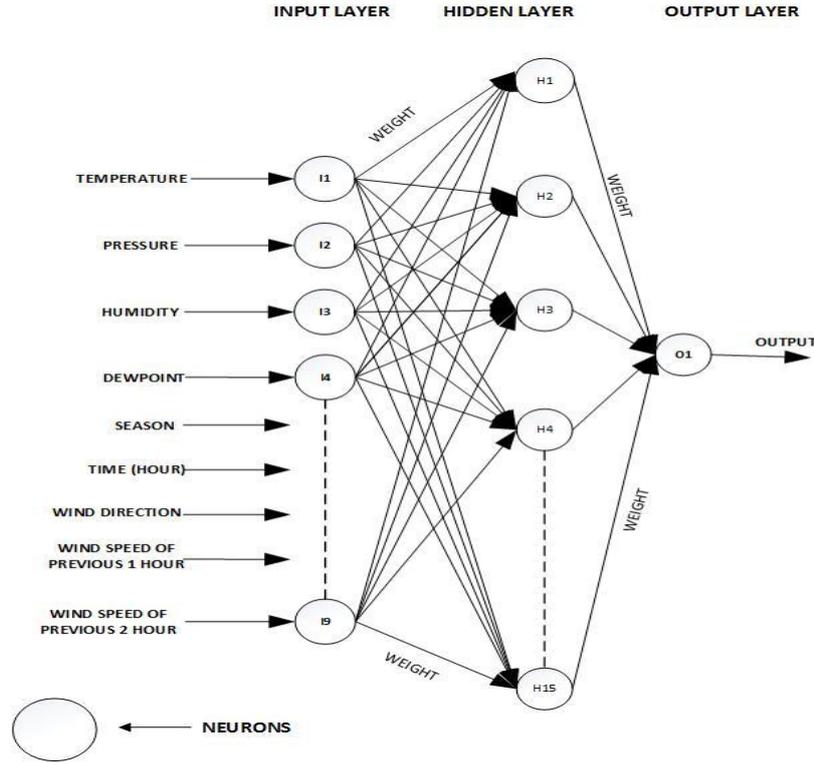


Fig. 1 Working model of an ANN

IV. Results

Above data is used to forecast wind speed for 48hr ahead time-scale using three different training algorithms separately. Results of each algorithm is discussed below for various hidden layer neurons, and finally various detailed performance and output plots are shown for the state at which best output is obtained.

4.1 Levenberg–Marquardt (LM)

The following study is done using Matlab Environment. In Matlab, the command used for training network using Levenberg-Marquardt backpropagation algorithm is “trainlm”.

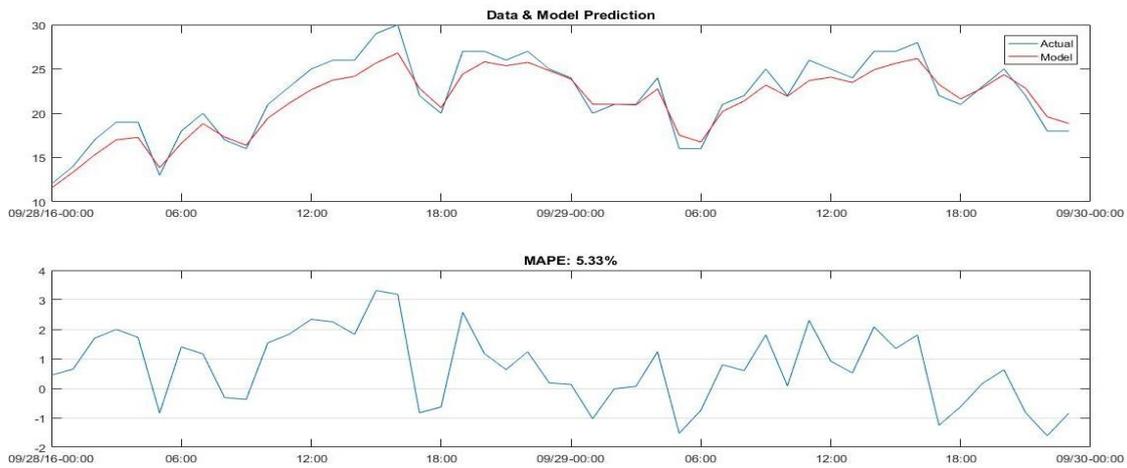


Fig. 2: Comparison between the predicted and actual wind speed employing the proposed model using LM training.

Mean Absolute Percent Error (MAPE): 5.67
 Mean Absolute Error (MAE): 1.2964
 Mean Square Error (MSE): 15.6
 Overall Regression: 0.85714
 Number of Epochs: 42

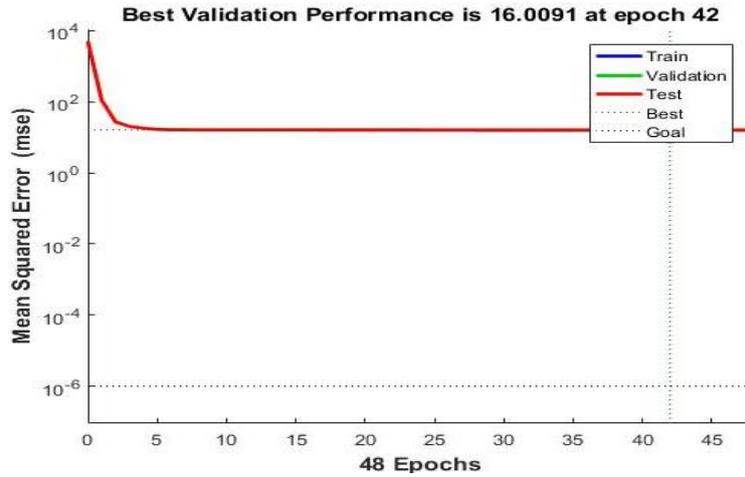


Fig. 3: Computed MSE value using the proposed LM trained Neural Network.

4.2 Scaled Conjugate Gradient (SCG)

The following study is done using Matlab Environment. In Matlab, the command used for training network using Scaled Conjugate Gradient (SCG) backpropagation algorithm is “trainscg”.

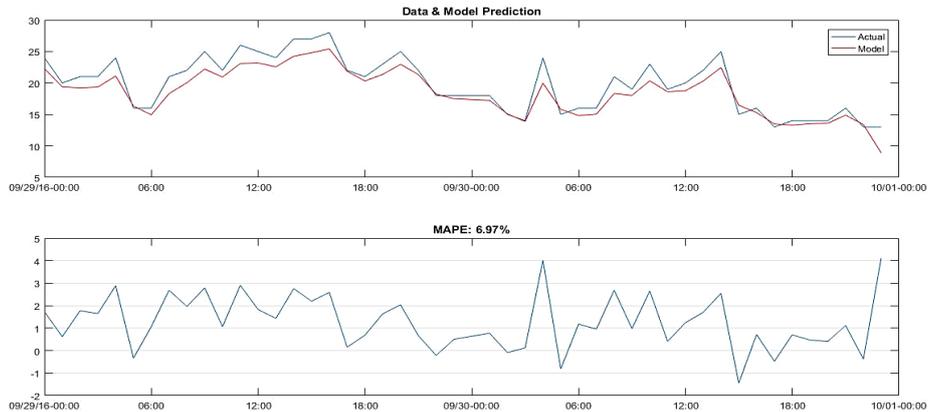


Fig. 4: Comparison between the predicted and actual training wind speed employing the proposed model using SCG training.

Mean Absolute Percent Error (MAPE): 6.97

Mean Absolute Error (MAE): 1.1582

Mean Square Error (MSE): 16.6

Overall Regression: 0.85042

Number of Epochs: 130

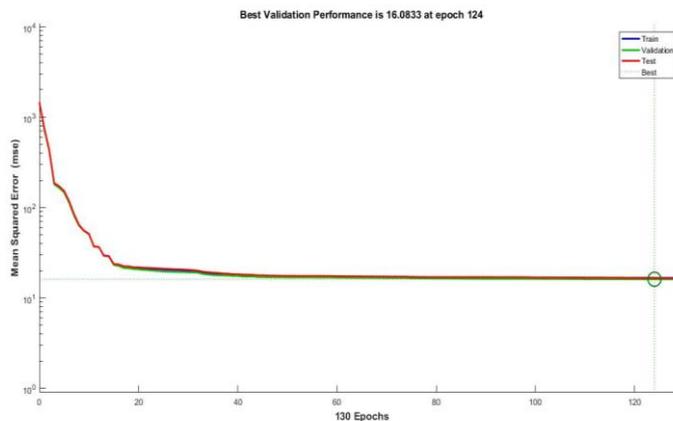


Fig. 5: Computed MSE value using the proposed SCG Trained Neural Network.

4.3 BAYESIAN REGULARIZATION (BR)

The following study is done using Matlab Environment. In Matlab, the command used for training network using Bayesian Regularization (BR) backpropagation algorithm is “trainbr”.

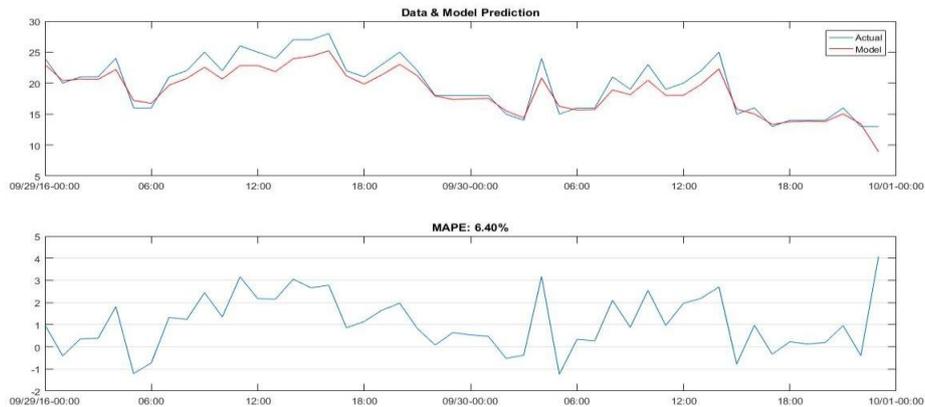


Fig. 6: Comparison between the predicted and actual wind speed employing the proposed model using BR training.

Mean Absolute Percent Error (MAPE): 6.40
 Mean Absolute Error (MAE): 1.3199
 Mean Square Error (MSE): 15.9
 Overall Regression: 0.85766
 Number of Epochs: 504

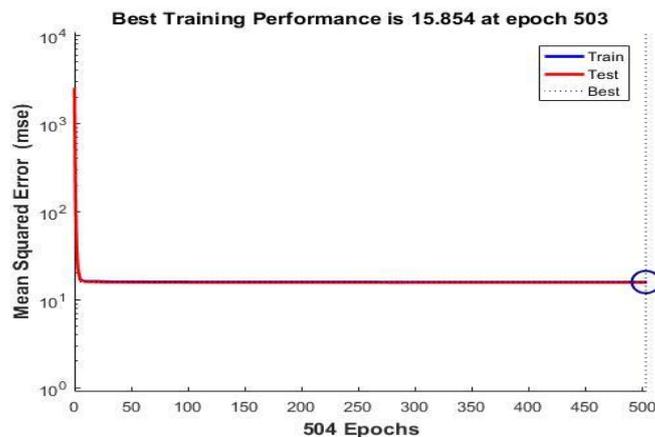


Fig. 7: Computed MSE value using the proposed BR trained model

V. Conclusion

In the present project, an attempt is made to forecast wind speed using three ANN models by utilizing historical weather data to train ANN model. The model developed is able to predict wind speed for next 48 time steps. Obtained results are summarized below:

- The developed LM model with the configuration of 9-15-1 gives the best result with MAPE =5.67 % and corresponding MAE = 1.2964 and MSE= 15.6 after 42 epochs.
- The developed SCG model with the configuration of 9-17-1 gives the best result with MAPE = 6.97 % and corresponding MAE = 1.1582 and MSE= 16.6 after 130 epochs.
- The developed BR model with the configuration of 9-13-1 gives the best result with MAPE = 6.40 % and corresponding MAE = 1.3199 and MSE= 15.9 after 504 epochs.

Out of the three techniques LM has least error hence it will give the most accurate forecasting also it has least epochs hence it has good convergence speed.

After analyzing the results, it has been found that forecasted data is very similar to actual measured data with an MAPE of 5-7 % which is a great improvement and hence ANN can be used to predict the generation of electricity for short term in wind power plant.

References

- [1]. P. Jennings, New directions in renewable energy education, *Renew. Energy* 34, pp. 435–439, 2009.
- [2]. Renewables 2016 Global Status Report, 2016.
- [3]. Department of renewable Energy Resources, Government of India, 2016.
- [4]. Fazelpour, F., Tarashkar, N. & Rosen, M.A, Short-Term Wind Speed Forecasting Using Artificial Neural Networks For Tehran, Iran, *International Journal Of Energy And Environmental Engineering (2016)*, Issue 4, pp. 377–390, 2016.
- [5]. Melam Bhaskar, Amit Jain, and N. Venkata Srinath, Wind Speed Forecasting: Present Status, *International Conference on Power System Technology*, 2010.
- [6]. Ulas Eminoglu, A New Model for Output Power Calculation of Variable-Speed Wind Turbine Systems, *Intl Conference on Optimization of Electrical & Electronic Equipment (OPTIM)*, pp. 141 – 146, 2015.
- [7]. R. Hyndman, and A. Koehler, Another look at measures of forecast accuracy, *International Journal of Forecasting* 22, pp. 679 – 688, 2005.
- [8]. Alexiadis, M.C., Dokopoulos, P.S., Sahasamanoglou, H.S., Manousaridis, I.M.: Short-term forecasting of wind speed and related electrical power. *Solar Energy Volume 63*, Pages 61–68, 1998.
- [9]. Giorgi, M.G.D., Ficarella, A., Tarantino, M. “Error analysis of short term wind power prediction models, *Appl. Energy* 88, pp. 1298–1311, 2011.
- [10]. P. M. Fonte, Gonçalo Xufre Silva, J. C. Quadrado, Wind Speed Prediction using Artificial Neural Networks, *6th WSEAS Int. Conf. on Neural Networks, Lisbon, Portugal, June 16-18*, pp. 134-139, 2005.
- [11]. D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, Vol. 11, pp. 431–441, 1963.
- [12]. K. Levenberg, A method for the solution of certain problems in least squares, *Quart. Appl. Mach.*, vol. 2, pp. 164–168, 1944.
- [13]. A Method of Accelerating Neural Network Learning, *Neural Processing Letters, Springer*, pp. 163–169, 2005.
- [14]. M. T. Hagan and M. B. Menhaj, Training feedforward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [15]. Bogdan M. Wilamowski and Hao Yu, Improved Computation for Levenberg–Marquardt Training, *IEEE Transactions on Neural Networks*, Vol. 21, No. 6, 2010.
- [16]. Ozgur Kisi; Erdal Uncuoglu, Comparison of three backpropagation training algorithms for two case studies, *Indian Journal of Engineering & Materials Sciences*, Vol. 12, pp. 434-421, 2005.
- [17]. E.M. Johansson, F.U. Dowl, and D.M. Goodman, Backpropagation Learning for Multilayer Feed-Forward Neural Networks using The Conjugate Gradient Method, *International Journal of Neural Systems*, vol. 02, 1991.
- [18]. Martin Foddslette Møller, A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks, Volume 6*, pp. 525-533, 1993.
- [19]. Mackay, D.J.C., Bayesian interpolation, *Neural Computation*, Vol. 4, pp 415-447, (1992).
- [20]. Zhao Yue; Zhao Songzheng; Liu Tianshi, Bayesian regularization BP Neural Network model for predicting oil-gas drilling cost, *Business Management and Electronic Information (BMEI), 2011 International Conference on 13-15, vol.2*, pp. 483-487, 2011.
- [21]. M. S. Miranda, and R. W. Dunn, One-hour-ahead wind speed prediction using a Bayesian methodology, *IEEE Power Engineering Society General Meeting*, pp. 1-6, 2006.
- [22]. Kanna Bhaskar and S.N. Singh, AWNN - Assisted Wind Power Forecasting using Feed-Forward Neural Network, *IEEE Transactions on Sustainable Energy, Volume 3*, pp. 306-315, 2012.
- [23]. Sourabh Mehto, Rahul Agarwal, Manish Shah, “A Comparative Study Based on PI and Fuzzy Control of Inverter fed Induction Motor Drive”, *International Journal of Research (IJR) Vol-2, Issue-2 February 2015 ISSN 2348-6848*, pp.75-79.
- [24]. Juhi Nagpal, Rahul Agarwal & Manish Shah, “A Comparative Study on Different Speed Control Methods of D.C. Drives for Electric Vehicle”, *International Journal of Research (IJR) e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 07, July 2015*, pp. 1-6.
- [25]. Khushboo Hardia, Neha Maithil & Manish Shah, “Particle Swarm Optimization Applied to Economic Load Dispatch Problem”, *International Journal of Research (IJR) e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 09, September 2015*, pp. 834-843.
- [26]. Jyoti verma, Manish Shah, “Simulation and Analysis of H-bridge Cascade Multi level Inverter Fed Induction Motor Drive for Industrial Applications”, *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering Vol. 4, Issue 3, March 2016*, pp. 17-19.
- [27]. Manish Shah, Rahul Agrawal, “A Review on Classical and Modern Techniques with Decision Making Tools for Load Forecasting”, *International Journal of Emerging Trends in Engineering and Development Issue 3, Vol.6 (November 2013)*, pp. 174-184.
- [28]. Ritu Bhateja, Manish Shah, A Proposed Method for Smooth Speed Control of Indirect Vector Control Inverter-Fed Induction Motor Drive using Boost Converter Topology, *International Journal of Emerging Trends in Engineering and Development Issue 6, Vol. 5*, pp. 1-10, 2016.
- [29]. Sudeep Mohaney, Sourabh Mehto, Manish Shah, Modeling and Optimization of Elevator Group Control System for High Rise Commercial Building”, *IEEE International Conference on Computer, Communication and Control (IC4-2015)*.
- [30]. Sudeep Mohaney, Manish Shah, Emerging Trends in Vertical Elevating System, *International Journal of Engineering and Management Research Vol.-5, Issue-1*, pp. 51-56, 2015.
- [31]. Khushboo Nagar, Manish Shah, A Review on Different ANN Based Fault Detection Techniques for HVDC Systems, *MIEEE, MIE, MISLE, International Journal of Innovative Research in Engineering & Management (IJIREM), Issue-6, 2016*.