Employing Data Analytics and Machine Learning for Analyzing and Forecasting Wind Speed Data

Sakshi Patel[1], Prof. Pallavi Bagde[2]
Department of CSE, SDBCT, Indore[1,2]

***Abstract*— Big data analytics is being used for various applications presently, out of which statistical analysis for renewable energy is of great importance. Out of all present renewable sources of energy, wind energy 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 and well as future long term plans. This study presents a comparative study of a Wind speed forecasting model using Artificial Neural Networks (ANN) with back propagation. Here an attempt is made to forecast Wind Speed using ANN with back propagation method 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) and Mean Absolute Percentage Error (MAPE).**

Keywords: Big Data Analytics, Wind Speed Prediction, ANN, back propagation, Multi-layer neural network.

I. INTRODUCTION

Currently every country in the world is focusing in increasing power generation through renewable sources in order to tackle high carbon emission while committing energy security [1]. Although in 2015, renewable energy shares only 19.2% 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,

$$P = \frac{1}{2} \rho \, \pi \, R^2 \, v^3 C_{P\_Max} \qquad (1)$$

Where v stands for wind speed (m/s), R is the radius of the rotor (m), ρ is the air density (kg/m^3), and C_{P_Max} 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 of Station Name: Austin Bergstrom AP, County: Travis State: Texas is utilized as input in algorithm development. This data is then fed to our proposed ANN as training and testing data after wavelet transform. 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 Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

Mean absolute error (MAE) is used to measure how close predicted speeds are to the actual speeds. The mean absolute error is given by:

$$MAE = \frac{1}{N}\sum_{t=1}^{N} |A_t - \hat{A}_t| = \frac{1}{N}\sum_{t=1}^{N} |e_t| \qquad (2)$$

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:

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

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 mimic the learning processes of the brain to discover the relations between the variables of a system. They process input data information to learn and obtain knowledge for forecasting or classifying patterns etc. type of work. ANN consists of number of simple processing elements called neurons. All information processing is done within this neuron only. 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].

Back Propagation Algorithm:

Reducing error function is the main reason to use this algorithm. Levenberg-Marquardt algorithm [11] [12] 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.
Levenberg-Marquardt algorithm is fast [13] and has stable convergence.
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

$$H = {J_x}^T J_x \quad (4)$$
$$g = {J_x}^T e \quad (5)$$

Where $\boldsymbol{J_k}$ 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,

$$\boldsymbol{W_{k+1}} = \boldsymbol{W_k} - [{\boldsymbol{J_K}}^T \boldsymbol{J_k} + \mu \boldsymbol{I}]^{-1} {\boldsymbol{J_K}}^T \boldsymbol{e_k} \quad (6)$$

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 β=l0. 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.

The following table depicts the comparison of three algorithms:

Algorithm	**Rules**	**Convergence**
Gradient Newton algorithm	$\boldsymbol{W_{k+1}} = \boldsymbol{W_k} - \boldsymbol{\alpha g_x}, \quad \boldsymbol{\alpha} = \frac{\mathbf{1}}{\mu}$	Stable, slow
Gauss–Newton algorithm	$\boldsymbol{W_{k+1}} = \boldsymbol{W_k} - [{\boldsymbol{J_K}}^T \boldsymbol{J_k}]^{-1} {\boldsymbol{J_K}}^T \boldsymbol{e_k}$	Unstable, fast
Levenberg –Marquardt (LM) algorithms	$\boldsymbol{W_{k+1}} = \boldsymbol{W_k} - [\boldsymbol{J^T}_{\boldsymbol{k}} \boldsymbol{J_k} + \mu \boldsymbol{I}]^{-1} \boldsymbol{J_k e_k}$	Stable, fast

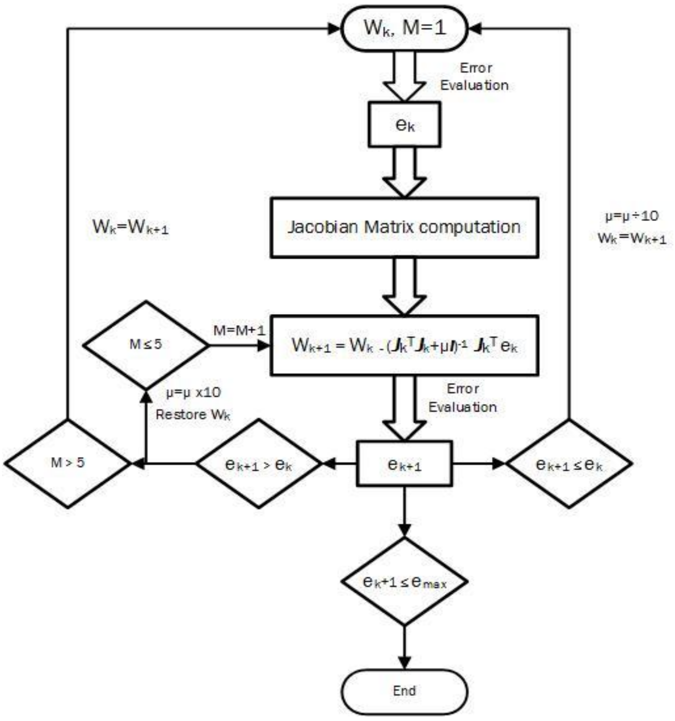

Fig. 1: Block diagram for training using Levenberg–Marquardt algorithm

Wavelet Transform for Time series problem:

By means of wavelet transform a time series can be decomposed into a time dependent sum of frequency components. As a result, we are able to capture seasonality's with time-varying period and intensity, which nourishes the belief that incorporating the wavelet transform in existing forecasting methods can improve their quality. The article aims to verify this by comparing the power of classical and wavelet based techniques on the basis of four-time series, each of them having individual characteristics. Depending on the data's characteristics and on the forecasting horizon we either favor a denoising step plus an ANN forecast or a multiscale wavelet decomposition plus an ANN forecast for each of the frequency components.

Now that we know some situations when wavelet analysis is useful, it is worthwhile asking the questions "What is wavelet analysis?" A wavelet is a waveform of effect, having a limited duration that has an average value of zero. Comparing wavelets with sine waves, which are the basis of Fourier analysis yields that Sinusoids do not have a limited duration, i.e. they extend from minus infinity to plus infinity, and while sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric as seen from the figure below:

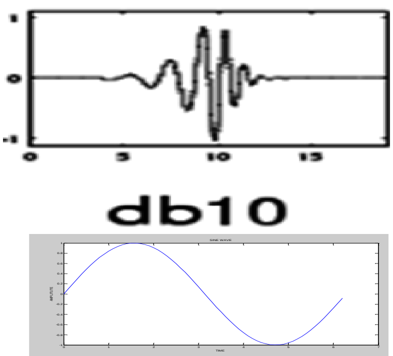

Fig. 2: (a) db-10 Wavelet (b) Sinewave

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet. Just looking at pictures of wavelets and sine waves, we can see intuitively that signals with sharp changes might be better analyzed with an irregular wavelet than with a smooth sinusoid, just as some foods are better handled with a fork than a spoon. It also makes sense that local features can be described better with wavelets, which have local extent.

Discrete Wavelet Transform (DWT) The disadvantage of the continuous wavelet transform lies in its computational complexity and redundancy. In order to solve these problems, the discrete wavelet transform is introduced. Unlike CWT, the DWT decomposes the signal into mutually orthogonal set of wavelets. The discrete wavelet is defined as:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}}\psi\left(\frac{t - k\,\tau_0^k s_0^j}{s_0^j}\right) \tag{6}$$

where j and k are integers, $s_0 > 1$ is a fixed dilation step and the translation factor τ_0 depends on the dilation step. The scaling function and the wavelet function of DWT are defined as:

$$\phi(2^j t) = \sum_{i=1}^{k} h_{j+1}(k)\,\phi(2^{j+1}t - k)$$

$$\psi(2^j t) = \sum_{i=1}^{k} g_{j+1}(k)\,\phi(2^{j+1}t - k)$$

And then, a signal f(t) can be written as:

$$f(t) = \sum_{i=1}^{k} \lambda_{j-1}(k)\,\phi(2^{j-1}t - k) + \sum_{i=1}^{k} \nu_{j-1}(k)\,\phi(2^{j-1}t - k)$$

The discrete wavelet transform can be done by using the filter bank scheme developed. Figure 3.1 shows a two-channel filter bank scheme for DWT.

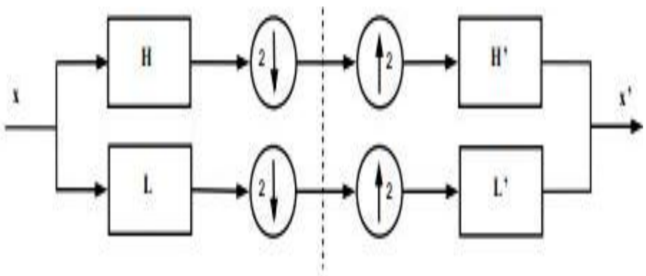

Fig. 3: Filter Bank Scheme for DWT

In the figure, H, L, and H', L' are the high-pass and low-pass filters for wavelet decomposition and reconstruction respectively. In the decomposition phase, the low-pass filter removes the higher frequency components of the signal and high pass filter picks up the remaining parts. Then, the filtered signals are down sampled by two and the results are called approximation coefficients and detail coefficients. The reconstruction is just a reversed process of the decomposition and for perfect reconstruction filter banks, we have x = x'. A signal can be further decomposed by cascade algorithm as shown in Equation 5):

$$
\begin{aligned}
x(t) &= A1(t) + D1(t) \\
&= A2(t) + D2(t) + D1(t) \\
&= A3(t) + D3(t) + D2(t) + D1(t) \\
&= An(t) + Dn(t) + Dn_{-1}(t) + \ldots\ldots + D1(t) \qquad (7)
\end{aligned}
$$

where Dn(t) and An(t) are the detail and the approximation coefficients at level n respectively. Fig 4 illustrates the corresponding wavelet decomposition tree.

A significant potential problem with the DWT is that it is a shift variant transform. Shift-variance is a phenomenon of not necessarily matching the shift of the one-level DWT with the one-level DT of the same shift of a data sequence. Due to shift-variance, small shifts in the input waveform. Wavelet Transform cause large changes in the wavelet coefficients and this makes the DWT suitable for this study because we relate the information at a given time point at the different scales.

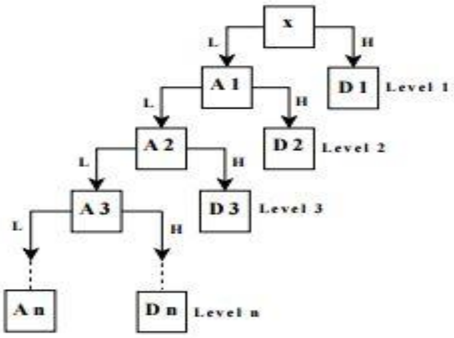

Fig. 4: Wavelet Decomposition Tree

III. DATA INPUTS

Following inputs elements are taken for forecasting wind speed,

- Temperature
- Pressure
- Humidity
- Dew Point
- Time (hour)
- Date
- Wind speed of previous one hour
- Wind speed of previous two hours

The Neural Network in present study consist of three layers. The first one is Input layer (consist of 9 Neutrons 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.

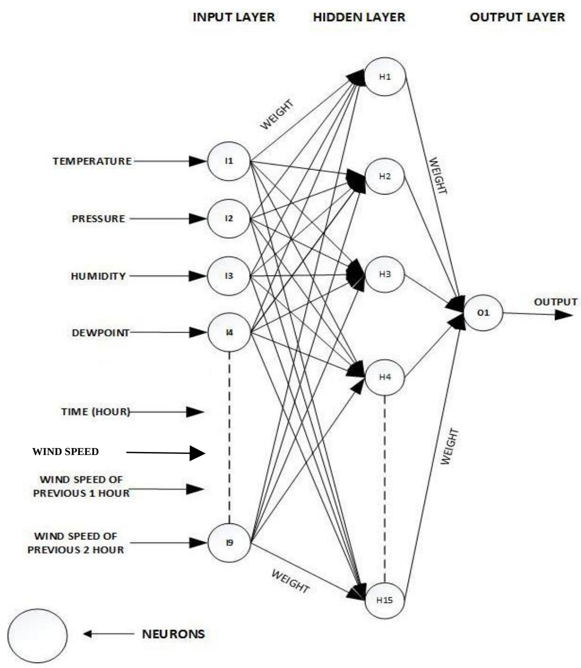

Fig. 5: Working model of an ANN

IV. EXPERIMENTAL RESULTS

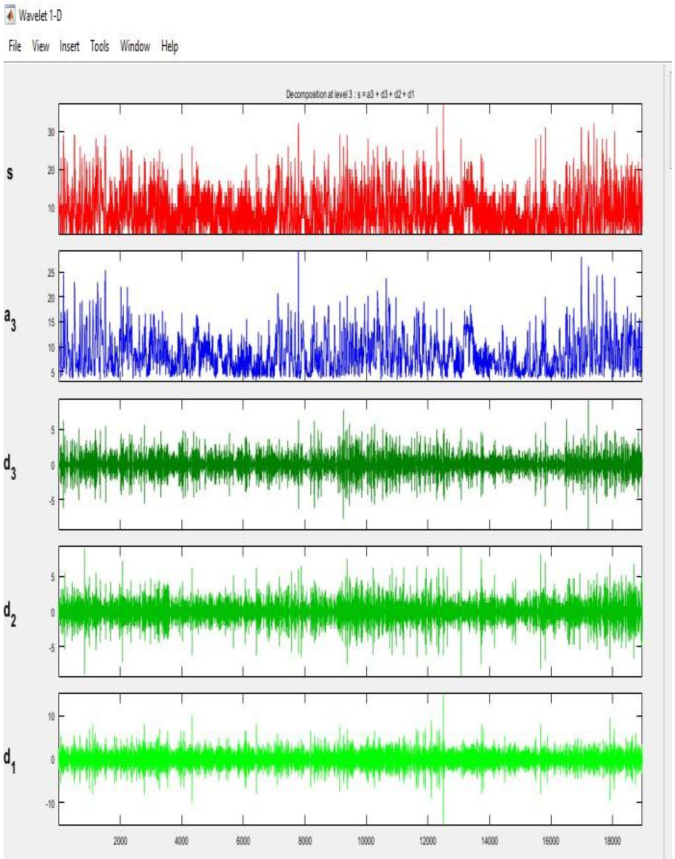

Fig. 6: Wavelet decomposition up to level 3 of wind speed.

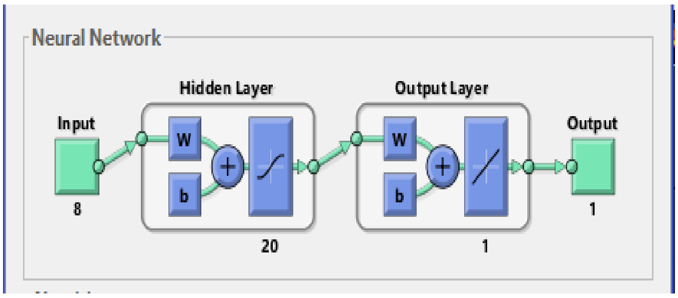

Figure 7: Graphical Diagram of The Proposed Neural Network for LM Training.

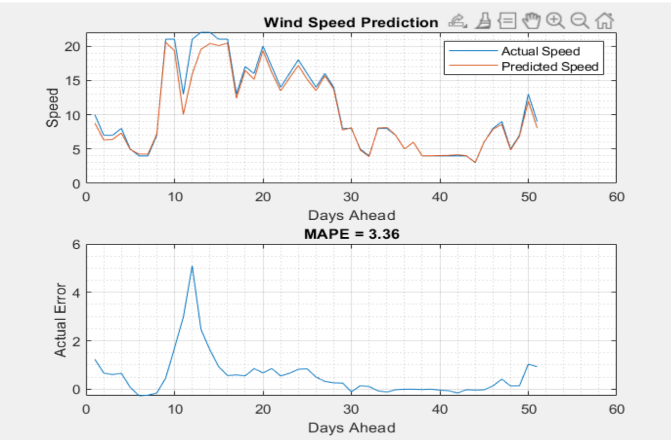

Fig 8: Comparison between the predicted and actual wind speed employing the proposed model using back propagation

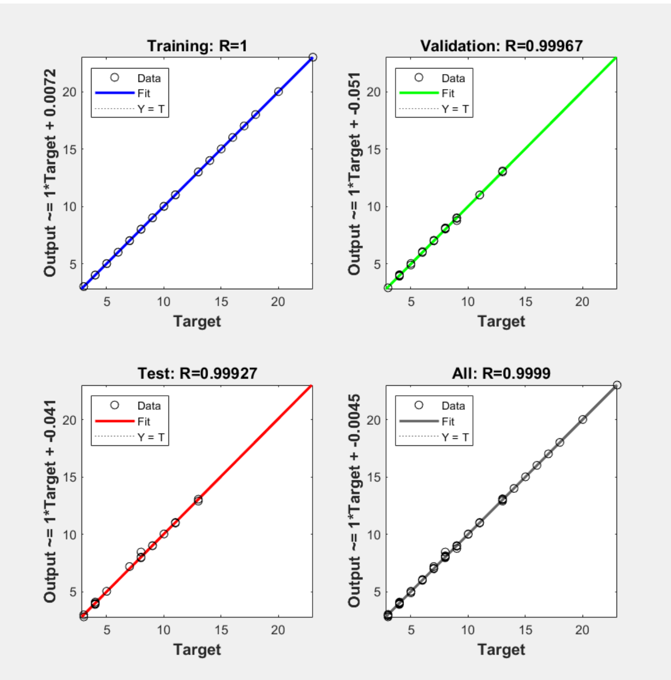

Fig 9: Regression plot during training, testing & validation for back propagation training algorithm.

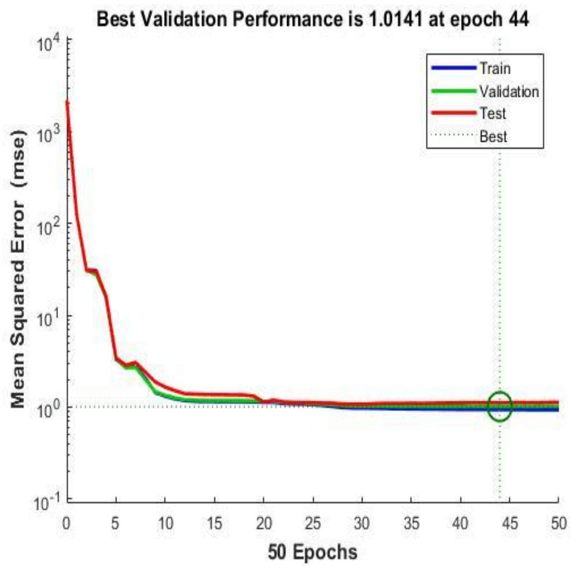

Fig. 9: Neural network performance during training, testing & validation

TABLE 1. Summry of Results

S.No.	Parameter	Value
1	Statistical Model	ANN
2	Architecture	Back Propagation
3	Pre-Processing	DWT
4	Mean Squared Error (MSE)	1.0141
5	Mean Absolute Percentage Error	3.36%
6	Regression (Overall)	0.99
7	Iterations	41

Conclusion: Wind speed forecasting is essential for various reasons, particularly in the context of renewable energy, environmental monitoring, and operational planning. In energy markets, accurate forecasting is vital for making informed decisions about buying and selling electricity. It allows market participants to anticipate changes in supply and demand, optimize energy trading strategies, and manage risks. This paper presents a back propagation based wind speed forecasting model which attains low MAPE and MSE at moderate iterations to convergence.

REFRENCES

[1] S. Liang, L. Nguyen and F. Jin, "A Multi-variable Stacked Long-Short Term Memory Network for Wind Speed Forecasting," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4561-4564.
[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). December 2016, Volume 7, Issue 4, pp 377–390 (spinger)
[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), 2015. Publication Year: 2015, Page(s):141 – 146.
[7] Hyndman, R. and Koehler A. (2005). "Another look at measures of forecast accuracy", International Journal of Forecasting 22 (2006), 679 – 688.
[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, Issue 1, July 1998, Pages 61–68.
[9] Giorgi, M.G.D., Ficarella, A., Tarantino, M. "Error analysis of short term wind power prediction models". Appl. Energy 88, 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, 2005, pp134-139.
[11] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," SIAM J. Appl. Math., 1963, Vol. 11, pp. 431–441
[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 (2005) 22:163–169
[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, Nov. 1994.
[15] Bogdan M. Wilamowski and Hao Yu, "Improved Computation for Levenberg–Marquardt Training", IEEE Transactions on Neural Networks, Vol. 21, No. 6, June 2010.
[16] X Wang, X Li, J Su – Energy, "Distribution drift-adaptive short-term wind speed forecasting", Energy, Elsevier 2023, vol.273, art. 127209.
[17] E Zhao, S Sun, S Wang, "New developments in wind energy forecasting with artificial intelligence and big data: A scientometric insight", Data Science and Management, Elsevier 2022, vol.5, no.2., pp. 84-95.