Forecasting of Wind Speed Using Artificial Neural Networks

Ramesh Babu N. 1, Arulmozhivarman P. 2

Abstract – Wind speed forecast is essential in wind energy conversion system and may fail to operate power plant at non-optimal region if not properly forecasted. This paper focuses the short term wind speed forecasting using conventional statistical method and artificial neural networks such as back propagation network (BPN), generalized regression neural network (GRNN) and radial basis function networks (RBFN). The developed algorithms and networks are trained and tested for wind speed data which are measured at an interval of 15 minutes. In this paper we compared the performance of RBFN and other networks for effective wind speed forecasting.

Keywords: Wind Speed Forecast, BPN, GRNN, RBFN, ARMA

I. Introduction

In the recent years, research interest on the renewable energy based power production [1] had been increased manifold due to the exhausting fossil fuels and the pollution emitted by them. Among the renewable energy resources wind energy is the one through which the power production cost is less [2]. The wind speed forecast for the wind energy industry is essential due to the following reasons: (i) wind farms unit maintenance (ii) to maintain optimality between conventional renewable energy parks (iii) for electricity bidding (iv) to schedule the power generators (v) to plan and schedule energy reserves and storages. Basically wind speed is a time series data measured at different instants of time. Based on the time duration of forecast it has been classified as short term (few seconds to minutes), medium term (few minutes to hours) and long term forecasting (few hours to days). It is very difficult to forecast the wind speed accurately due to its randomness characteristics. At present, many research papers discuss various methods of wind speed forecasting.

Majority of statistical methods assume the time series is linear and stationary. In convention the series is assumed to have zero mean. Autoregressive model and Moving Average models are most widely used for forecasting [3]. Autoregressive Moving Average model (ARMA) is the forecasting model which gives more accurate results in very short time estimation.

The literature published to date shows various methods to predict and forecast the wind speed at different time scales. The short horizon data driven algorithms are presented [4] for wind speed prediction. Artificial Neural Networks are best suitable for prediction and forecast of wind speed data [5]-[8]. Other methods based on numerical weather prediction are also used for wind speed forecast [9], [10].

The recent advancements in forecast and prediction of wind data has been reviewed thoroughly [11] and suggestions for benchmarking the wind forecast problem were proposed. It has been concluded from the literature there is a need of improvement in the approaches attempted for forecast the short term wind speed data.

This paper developed on that background and here we propose Radial Basis Function Network (RBFN) based model for wind speed forecast and the results are compared with other models chosen from the literature to validate the performance improvement.

II. Methodology

II.1. ARMA Model

ARMA model established by Box and Jenkins have been used widely for time-series forecasting applications. This model uses Autoregressive model followed by...
The Moving Average model. The ARMA (p, q) model [12] is formulated by the characteristic equation:

\[ X_t = C + \sum_{j=1}^{p} \varphi_j X_{t-j} + \sum_{k=1}^{q} \beta_k \varepsilon_{t-k} + \varepsilon_t \]  

(1)

where \( C \) is a constant term, \( \varphi_j \) is the \( j \)th autoregressive coefficient, \( \beta_k \) is the \( k \)th moving average coefficient, \( \varepsilon_t \) is the error term at time \( t \), \( \varepsilon_{t-k} \) is the random error (White Gaussian) at previous instants, \( p \) and \( q \) are order of autoregressive and moving average terms respectively.

The AR operation (IIR filter) retains the information of historic dependence between future and past values. The MA operation (FIR filter) retains the information of successive errors, perturbations that influence the final result. This model formulation has three basic steps: modeling, characterization and forecasting. Modeling involves capturing of long term behavior features accurately, characterization is to determine the amount of randomness and forecasting is predicting the short term system evolution.

The performance evaluation is verified using Akaike’s information criterion (AIC) or Bayesian information criterion (BIC) for the effectiveness. After the model identification, lowest residual of the model is generated by using either Yule-Walker Estimation or Maximum Likelihood Estimation.

II.2. ANN-BPN Model

Artificial neural network is widely used in time forecasting models due to its characteristics of extreme computational power, massive parallelism, and fault tolerance and it is more efficient through learning without enormous programming.

Neural networks not only learn the smooth prediction function but also are trained to enumerate unexpected short term regularities in a time series data. There are various ANN model structures available in literature, but back propagation network has been the most widely used model [13], [14].

The basic architecture of BPN is shown in Fig. 1. The output of the BPN network can be expressed as:

\[ y_j = f \left( \sum_i w_{ij} x_i \right) \]  

(2)

where, \( y_j \) is the output of node \( j \), \( f(.) \) is the transfer function, \( w_{ij} \) is the network connection between node \( j \) and node \( i \) and \( x_i \) is the input signal from node \( i \) to node \( j \). However, the determination of number of input and hidden nodes of the architecture is not completely identified. Different architectures use different nodes and hidden layers.

But for optimal solution we can choose the number of hidden layers empirically as:

\[ N = \left( \sqrt{a+1} \right) - 10 \]  

(3)

where, \( N \) is number of hidden layers, \( a \) is number of input neurons.

Also to avoid the problem of over fitting in the network, the observed data is divided into three parts as training, validation and testing for evaluating the forecasting performance.

![BPN Architecture](image)

**Fig. 1. BPN Architecture**

II.3. ANN-GRNN Model

Generalized regression neural network is closely related to probabilistic neural network and it is a better alternative to the BPN model. Unlike, Probabilistic Neural network (PNN) which is used for mapping, GRNN is used for estimation.

From computation point of view, GRNN is based on the estimation of probability density function from observed samples using Parzen window estimation. The sample estimation is expressed as:

\[ y(x) = E[y|x] = \frac{\sum_{i=1}^{n} y_i \exp(-d_i)}{\sum_{i=1}^{n} \exp(-d_i)} \]  

(4)

where, \( x, y \) are the measured values of input, \( X \) and output, \( Y \) respectively and \( d_i \) is the distance function between input vectors and centers recorded in pattern nodes given by:

\[ d_i = \frac{(x-x_i)^T (x-x_i)}{2\sigma^2} \]  

(5)

In the formulation of GRNN, all input variables and pattern nodes share a single common \( \sigma \) which is the smoothing factor or widening factor of the kernel. Choosing independent \( \sigma \) for variables improves the accuracy of the result. The learning technique in GRNN is different from other neural network learning methods.
In this learning is done almost instantaneously once presented with training data rather than iterative method and hence learning is faster in GRNN. In addition, GRNN does not converge to local minima. The major disadvantage of this technique is the amount of computation required for the estimation of the signal from the trained data.

II.4. ANN-RBFN Model

Radial Basis Function Network (RBFN) is proposed by Powell M.J.D in 1985, which has a theoretical basis of interpolation theory for networks. RBFN is basically a feed forward network uses Gaussian activation functions to approximate the network. Because of slow convergence and possibility of falling into local minima is high in BPN, researchers pay more attention towards RBFN in the recent years.

The basic architecture of RBFN involves three layers with different roles. The input layer is a fan-out layer that connects network to the environment. The only hidden layer applies a nonlinear transformation from input to hidden space of high dimensions. The output layer is a linear one. The \( i \) th output of the network can be expressed as:

\[
y_i = \sum_{k=1}^{h} w_{ik} \left( \| x - c_k \| \right) \phi_k(x)
\]

where, \( x = [x_1, x_2, ..., x_n]^T \) is an input vector; \( n \) is the number of input nodes; \( c_k \) is the \( k \)th center node in the hidden layer; \( k = 1, 2, ..., h \), \( h \) is the number of hidden nodes. \( \| x - c_k \| \) denotes the Euclidean distance between \( c_k \) and the input vector \( x \). \( w_{ik} \) is the weight between hidden layers and output nodes. \( \phi_k(x) \) is a nonlinear function and it is radially symmetry. \( \phi_k(x) \) should be a Gaussian function expressed as:

\[
\phi_k(x) = \exp\left(-\frac{\| x - c_k \|^2}{\sigma_k^2}\right)
\]

The unique feature of RBFN lies in the process performed in hidden layer. The patterns in the input space forms clusters usually through k-means clustering.

If the centers of clusters are known then the distance from the cluster centre can be measured and the network from hidden to output layer becomes linear mapping. The area is radially symmetrical around the cluster centers.

The argument of the activation function of each hidden unit in an RBFN network computes the Euclidean norm between input and the center.

The activation function can be smoothened by properly choosing the width parameter \( \sigma_k \).

III. Results

The ARMA model was implemented for five different structures and their moving average percentage errors (MAPE) are listed in Table I. The results indicated that the most appropriate ARMA model to forecast the wind speed was ARMA (1, 1). After the model was chosen the forecast output of ARMA is shown in Fig. 2 compared with the actual data curve.

<table>
<thead>
<tr>
<th>Model</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMA (0,1)</td>
<td>3.8808</td>
</tr>
<tr>
<td>ARMA (1,0)</td>
<td>3.9111</td>
</tr>
<tr>
<td>ARMA (1,1)</td>
<td>3.8302</td>
</tr>
<tr>
<td>ARMA (2,1)</td>
<td>3.9584</td>
</tr>
<tr>
<td>ARMA (1,2)</td>
<td>3.8148</td>
</tr>
</tbody>
</table>

Feed forward back propagation neural network is trained for the given data to select the number of neurons in the hidden layer. During the training, mean square error (MSE) is considered as the performance function.

The simulation results are obtained for different neurons in hidden layer and shown in Table II. It is observed from Table II that MSE is less for 12 neurons with minimum convergence time for hidden layers. Hence 12 neurons are considered for BPN architecture. Also, various training algorithm for BPN network is chosen and performance is measured for 12 neurons in hidden layer. The training results are shown in Table III.

![Fig. 2. Measured vs Forecasted Output of ARMA Model](image-url)
From the Table III it is observed that Levenberg-Marquardt (LM) algorithm based BPN shows least MSE and hence it has been decided to choose BPN with LM algorithm and 12 neurons in hidden layers. The regression plot for the chosen architecture is shown in Fig. 3.

The simulation results of forecasted and actual wind speed for the chosen architecture model of BPN network and GRNN model is shown in the Fig. 4 and Fig. 5 respectively.

### TABLE III

**PERFORMANCE OF BPN WITH VARIOUS TRAINING ALGORITHMS**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Goal (epochs)</th>
<th>Performance (MSE)</th>
<th>Time (seconds)</th>
<th>Regression Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levenberg-Marquardt</td>
<td>1000</td>
<td>0.0256</td>
<td>79</td>
<td>0.99128</td>
</tr>
<tr>
<td>Gradient descent</td>
<td>1000</td>
<td>0.163</td>
<td>18</td>
<td>0.9518</td>
</tr>
<tr>
<td>Momentum</td>
<td>1000</td>
<td>0.08864</td>
<td>33</td>
<td>0.97554</td>
</tr>
<tr>
<td>Scaled conjugate gradient</td>
<td>1000</td>
<td>0.121</td>
<td>19</td>
<td>0.93779</td>
</tr>
</tbody>
</table>

### TABLE IV

**PERFORMANCE COMPARISON OF WIND SPEED FORECAST**

<table>
<thead>
<tr>
<th>Method</th>
<th>Simulation Time (s)</th>
<th>MSE</th>
<th>MAE</th>
<th>Regression</th>
<th>SSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPN</td>
<td>65</td>
<td>0.1954</td>
<td>0.3028</td>
<td>0.9934</td>
<td>58.43</td>
</tr>
<tr>
<td>GRNN</td>
<td>40</td>
<td>0.0234</td>
<td>0.0418</td>
<td>0.9938</td>
<td>7.118</td>
</tr>
<tr>
<td>RBFN</td>
<td>65</td>
<td>0.0099</td>
<td>0.022</td>
<td>0.9923</td>
<td>2.973</td>
</tr>
</tbody>
</table>

### IV. Conclusion

In this paper, various ARMA models for forecasting short-term wind speed has been developed and tested. The results indicated that the ARMA (1, 1) was the best fit model for the chosen data set.

Three feed forward neural networks (BPN, GRNN, and RBFN) are employed in this research to forecast the wind speed.
Along with the wind speed other parameters like temperature, humidity and pressure are also considered to obtain the better forecast.

Extensive experimentation and comparison of these algorithms reveals its effectiveness on the results and it is shown that RBFN model outperform the other rival neural models in terms of performance measures of the wind speed forecast and improvement gained over other ANN models as well as the benchmark ARMA model.

Fig. 6. Measured vs Forecasted Output of RBFN Model

References


Authors’ information

Ramesh Babu N., Arulmozhivarman P.

Ramesh Babu N. was born in India in 1979. He received his bachelor’s degree in Electrical and Electronics Engineering in Bharathiar University, Tamilnadu, India in the year 2000 and received his master’s degree in Applied Electronics from Anna University, Tamilnadu, India in the year 2006. Currently he is pursuing Ph.D in VIT University, Tamilnadu, India. He has published several technical papers in national and international conferences and international journals. His current research includes Wind Speed Forecast and Optimal Control of Wind Energy Conversion System.

Arulmozhivarman P. was born in India in 1975. He received his Ph.D degree in the field of wavefront sensing and Adaptive optics from NIT Trichy, India in 2005 and he did both M.Sc., and B.Sc., Degree in Applied Physics(Instrumentation) in Bharathidasan University Trichy, and secured gold medal for both the programs. Currently he is working as Associate Professor in Signal processing & Image Processing Division under the School of Electronics Engineering, Vellore Institute of Technology University (VIT), Vellore, India. His research interest includes Biomedical Signal processing & Image Processing, Vision based surveillance system, Biometric detection system. Also he has undertaken DRDO Sponsored research projects in the area of remote sensing and image processing. Currently he is guiding 8 Ph.D candidates in the areas of Signal and Image processing and he has published more than 25 papers in National and International Journal. He served has one of the editor in several international conference proceedings and he is a reviewer of International Journal of Computer Engineering Research and IEEE Photonics Technology letters. Dr. P. Arulmozhivarman is a member of IEEE signal processing society.