

# Artificial Intelligence Techniques for Wind Power Prediction: A Case Study

C. Balakrishna Moorthy<sup>1\*</sup>, Ankur Agrawal<sup>2</sup> and M. K. Deshmukh<sup>1</sup>

<sup>1</sup>Department of EEE and E&I, BITS Pilani, K. K. Birla Goa Campus, Sancoale - 403726, Goa, India; cbkmoorthy@goa.bits-pilani.ac.in, mkd@goa.bits-pilani.ac.in

<sup>2</sup>Department of Mathematics, BITS Pilani, K. K. Birla Goa Campus, Sancoale - 403726, Goa, India; f2011400@goa.bits-pilani.ac.in

## Abstract

In this paper, the wind power is predicted using artificial intelligence techniques. The wind speed data as input and wind power data as output, measured at ten minutes interval for a period of eleven months are used for the study. The artificial intelligence techniques such as artificial neural network and genetic algorithm are used for prediction of wind power. The Genetic Algorithm (GA) and Back Propagation Algorithm (BPA) are used as learning algorithm in the artificial neural network. Different parameters such as learning rate, momentum coefficient and epochs are varied in the back propagation algorithm to obtain the best architecture in the neural network. Similarly, crossover fraction and elite count are varied along with number of generations in the genetic algorithm learning technique to select the best model. Furthermore, two hybrid methods such as combination of the two algorithms, namely BPA-GA and GA-BPA are also proposed for prediction of wind power. The performance of the four models are compared in terms of Mean Square Error (MSE). From the results, it is observed that genetic algorithm outperforms the rest in terms of accuracy in prediction.

**Keywords:** Artificial Neural Network, Back Propagation Algorithm, Genetic Algorithm, Prediction, Wind Power

## 1. Introduction

Among the various renewable-energy sources for electric power generation, wind energy is the fastest growing and most promising renewable energy source as it is proved to be economically viable. India ranks fifth in the installed wind power, explicitly the grid-connected wind power, with China, USA, Germany and Spain leading in net installed capacity<sup>1</sup>.

The energy available in the wind varies as the cube of the wind speed. So an understanding of the characteristics of the wind resource is important. The most prominent characteristic of the wind resource is its variability. The wind is highly variable, both geographically and temporally, but, as the wind speeds and direction vary constantly, the output of a wind farm varies

accordingly. The demand for energy at any time, either domestic or industrial is a fixed known quantity. Thus, to meet with this demand, planning beforehand is required. So efforts have been made to predict the behavior of wind and the corresponding electric power production. The prediction of wind power is necessary as wind is an intermittent source of energy. The prediction of wind power is an important tool for utilities to ensure grid stability and a favorable trading performance in the electricity market<sup>27</sup>.

Many statistical models have been used to predict the wind power<sup>41</sup>. These methods include ARIMA<sup>32,45</sup>, Mean Variance Estimation (MVE) technique<sup>2</sup>, principal component analysis<sup>3</sup>, ANFIS<sup>4</sup>, Fuzzy logic<sup>4</sup> etc.

Artificial Neural Networks (ANNs) have been used to predict the wind speed and energy<sup>27</sup>. Various

\*Author for correspondence

researchers used neural networks technique for wind speed prediction and compared its performance with an autoregressive model (ARIMA) and Persistence technique<sup>5</sup>. Fonte et al.<sup>6</sup> used them to predict the average hourly wind speeds by choosing the patterns set length. Perez et al.<sup>7</sup> used ANN with back propagation algorithm to predict the wind speed. Upadhyay et al.<sup>8</sup> also used a multi-layered feed-forward ANN trained by the resilient back propagation learning algorithm for hourly forecasting of wind speed in the region of Canada. Gong et al.<sup>9</sup> suggested a robust two-step methodology for accurate wind speed forecasting based on Bayesian combination algorithm and three neural network models, namely, Adaptive Linear Element Network (ADALINE), Back Propagation (BP) network and Radial Basis Function (RBF) network. Catalao et al.<sup>10</sup> proposed ANN in combination with wavelet transform for short-term wind power forecasting. Gnana et al.<sup>11</sup> proposed a hybrid model to predict wind speed based on model which integrates self organizing feature maps and multilayer perceptron network.

Genetic Algorithms (GA) is used to predict stock-price index<sup>12</sup>. GA is combined with the ANN for training the feed forward neural networks networks<sup>13</sup>. Diaz et al.<sup>16</sup> discuss the importance of the initial population on the genetic algorithm theoretically and states its impact on the GA model design. Rovea et al.<sup>17</sup> investigated the influence of the population size on the GA performance using the mathematical model of an E. coli fed-batch cultivation process. Che et al.<sup>18</sup> compare the back propagation and GAs using three different data sets. Xu et al.<sup>39</sup> compared back propagation method with GA for wind power prediction.

The work stated in the literature use the BPA and GA, but do not give insight into the parametric variations on the performances of these algorithms. Also, two combinations, namely BPA-GA and GA-BPA are proposed for prediction of wind power. The motivation here is to improve the performance of one algorithm by using the second algorithm. Finally, a new model to predict the wind power is developed.

## 2. Artificial Neural Network

ANN is an artificial intelligence technique, which tries to mimic the biological processing of the brain. ANN has been applied to engineering applications, various sciences including physics, chemistry and applied mathemat-

ics. Many researchers have used ANN to forecast stock price and other fields of engineering<sup>36</sup>. In general, there are different types of ANN namely, single layer network, multilayer network and recurrent networks. In this study, a multilayer network is used.

### 2.1 Multilayer Feed Forward Neural Network

In multilayer feed forward neural network, there are three layers, namely, input layer, hidden layer and output layer. In this study, wind speed data is considered as input neuron in the input layer. Deciding the number of neurons in the hidden layers is very important in the ANN architecture. Though these layers do not directly interact with the external environment, they have a tremendous influence on the final output. Using less number of neurons in the hidden layers will result in under fitting. Also using more number of neurons in the hidden layers may result in over fitting. Therefore some decisions must be taken to choose the number of neurons in the hidden layer. As such there is no theoretical limit on the number of hidden layers but typically they are just one or two. In this paper, a single hidden layer has been chosen. Also, the number of neurons in the hidden layer is varied from 4 to 10 to get optimal solution. The wind power is taken as output neuron in the output layer. A bias variable, value of which is 1, is added to the input and hidden layer. This is done to cover all linear functions possible and not just the ones passing through the origin. Suppose there are  $n$  inputs and  $m$  neurons in the hidden layer. The inputs for each of the  $m$  neurons will be sum of the product of weights and the inputs including the bias variable. The weights will be different for each neuron. They are chosen randomly and then updated by the learning algorithm. These weights act as an input for the activation function at each neuron. Activation function accounts for the non-linearity of the neural networks. The activation function used in this study is sigmoid function (1) at the hidden layer and hyperbolic tangent sigmoid function (2) at the output layer.

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (1)$$

$$\sigma(a) = \frac{2}{1 + \exp(-2a)} - 1 \quad (2)$$

The ANN network architecture with feed forward neural network is shown in Figure 1.

### 3. Learning Methods in ANN

ANN have three types of learning algorithms<sup>20,38</sup>. The classification of the learning algorithms in the ANN is shown in Figure 2.

#### 3.1 Supervised Learning

If the data available has the actual output data, and this data is used to compute errors between the predicted outputs and the output data labels, then the learning is supervised. The name supervised suggests that the learning is being corrected at all steps by the actual output.

#### 3.2 Reinforced Learning

In this type of learning, output data is not available, but some information is available about whether the learnt output is correct or incorrect. Rewards are given for correct outputs and penalty is deducted for the wrong ones.

#### 3.3 Unsupervised Learning

In this type of learning, actual output data labels are not available. The algorithm learns by itself under no 'supervision', and organizes itself. This is often referred to as self-organization or adaption. Networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network.

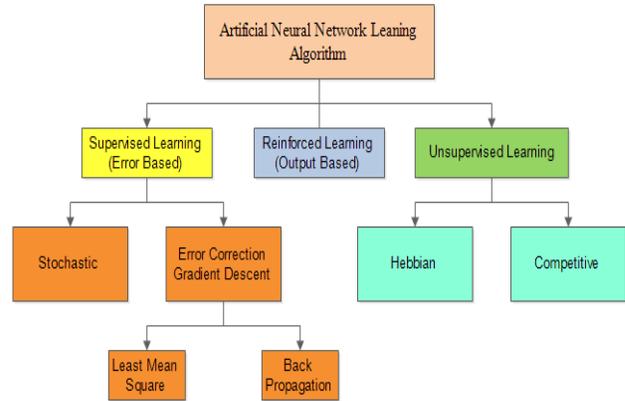


Figure 2. Classification of learning algorithms of the ANN<sup>20</sup>.

### 4. Back Propagation Algorithm (BPA)

BPA is a learning algorithm, which learns the weights of the feed forward neural network. The goal, as in any algorithm is to minimize the error. Given a training set comprising a set of input vectors  $X_n$ , where  $n = 1, \dots, N$ , together with a corresponding set of target vectors  $t_n$ , the error is given by Equation (3).

$$E(w) = \frac{1}{2} \sum_{n=1}^N \{y(X_n, w) - t_n\}^2 \tag{3}$$

This error is a function of the network weights  $w$ . The objective here is to find the weights such that the error is minimum. The minimum of a error function occurs when the gradient of that function is zero. At each step the weight vector is moved in the direction of the greatest rate of decrease of the error function. So this approach is known as gradient descent or steepest descent. The flow diagram of back propagation is shown in Figure 3.

The error at the output neuron is given by

$$\delta_k = y_k - t_k \tag{4}$$

Error at the hidden layer with the activation  $f$  is given by

$$\delta_j = \sigma'(a_j) \sum_k w_{kj} \delta_k \tag{5}$$

$$w_{ji}(\tau+1) = w_{ji}(\tau) - \eta \nabla E_n(w_{ji}(\tau)) \tag{6}$$

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \tag{7}$$

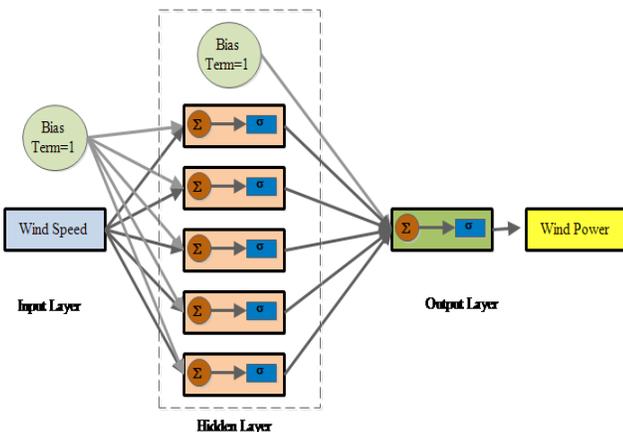


Figure 1. Developed neural network architecture.

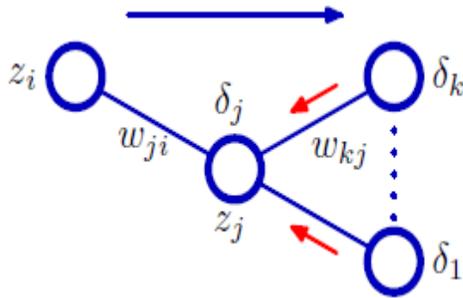


Figure 3. Backward propagation of error<sup>38</sup>.

where in (6) is known as the learning rate. Equation (6) gives the weights of the  $(+1)^{th}$  epoch in relation with the epoch.  $w_{ji}$  represents the weights between  $i^{th}$  and  $j^{th}$  layer,  $w_{kj}$  is the weights between the layer  $j$  and the output layer  $k$ .  $z_i$  in (9) is the output at the  $i^{th}$  layer after applying the activation function is the error at the output layer is the error at the hidden layer is the gradient,  $y_k$  is the predicted output and  $t_k$  is the target output.

### 4.1 Learning Rate

Learning rate controls how fast or how slow the algorithm reaches the zero gradient. Adjusting learning rate is important while using BPA. If the learning rate is too high, the algorithm can oscillate and become unstable. If it is too low, the algorithm may take long time to reach the converging point<sup>22</sup>. In this study, the value of learning rate such as 0.1, 0.5 and 0.9 are chosen.

### 4.2 Momentum Coefficient

Momentum coefficient also affect network convergence. The back propagation algorithm takes into account only the first order terms i.e., only the gradient. Hence, it can get stuck in a shallow local minimum. Momentum coefficient of 1 means that the network is insensitive to the gradient, and therefore gives poor results<sup>23</sup>. In this study, the value of momentum coefficient 0.1, 0.5 and 0.9 are varied to find the optimal solution in-terms of accuracy.

## 5. Genetic Algorithm (GA)

GA<sup>34</sup> is an algorithm that tries to mimic the process of evolution. It can be combined with the neural networks and be used to update its weights<sup>40</sup>. A random initial population which consists of the values of the variables

is considered initially. The population size of 200 is chosen in this study. Initially, 200 random values of the variable are chosen, which are the weights of the neural networks. The error is calculated for each variable and a few fittest individual are selected. The selection is based on a fitness function defined by (3). The individual with least error is selected. The next generation is made by the elite population either by mutation or by crossover. The flowchart for implementation of genetic algorithm is shown in Figure 4.

### 5.1 Elite Count

The number of individuals selected depends on a parameter called elite count. These few individual then generate a second generation of population. This is an important parameter which affects the performance of GA<sup>26</sup>. If the elite count is high, the GA will need more number of generations to converge. If it is too less, then the algorithm may give the value of only the fittest individual in the initial population. In this study, the elite count is varied as 0.10, 0.15 and 0.20.

### 5.2 Mutation

Mutation changes some characteristics (gene) of the variable vector of the individual to generate a new individual with a similar characteristics but not the variable vector. Amount of mutation is controlled by a mutation function. It has been taken as the default uniform function for this study.

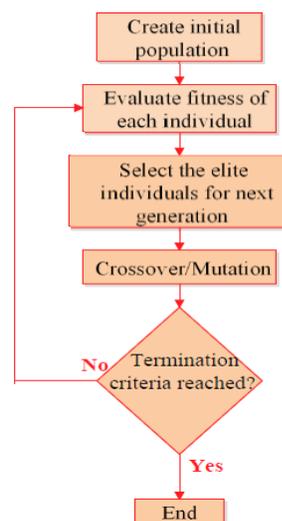


Figure 4. The working of Genetic Algorithm.

### 5.3 Crossover

Crossover takes two individuals in the elite, and takes part of the first individual, i.e., some values in the weight vector of the first, and the remaining values of the second individual to form a new individual. This happens stochastically and not deterministically<sup>25</sup>. The number of new individual created by crossover and mutation are controlled by a crossover fraction. The cross over fraction as 0.7, 0.8 and 0.9 is varied. The algorithm stops when the number of generations exceed a specified value or when the change in the average fitness value is less than the tolerance limit specified.

## 6. Data Description and Implementation

The wind speed and wind power data used in this study are obtained from a wind-farm, Tamil Nadu, India. The data is measured at a wind turbine having the rated capacity of 1500 kW, installed at a height of 85 m above ground level, recorded at 10 minutes interval. The total data consisting of 11 months spanning from 1<sup>st</sup> April, 2014 to 28<sup>th</sup> February, 2015. Thus, the total data points are 48096. The monthly variation of wind speed data at the site for a period of 11 months is shown in Figure 5.

The data are normalized to a range between 0 and 1 using (8).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (8)$$

Where,  $x_{norm}$  is the normalized data and  $x_{min}$ ,  $x_{max}$  are minimum and maximum values in the data to be normalized. Normalization is done so as to take a standardized range for the input. A standard range makes the training faster and more accurate<sup>21</sup>. For simulation, the data are divided into three sets: Training set, validation set and testing set.

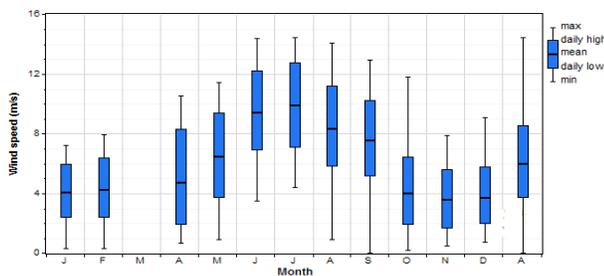


Figure 5. Monthly variation of wind speed at the site.

### 6.1 Training Set

It is the data set, which is used to train the neural network and fix the weights. The larger the training set the more comprehensive learning will occur. In this study, two training data sets are used. One with 80% of the data as the training data, i.e., 38477 data points. The other has 90% of the data points, i.e., 43286 data points. The error calculated during training is called training error.

### 6.2 Validation Set

The set defines the performance of a network. The lower the validation error the better the model is. For the 80% training data set, 10% of the data is used as the validation set, i.e., 4809 data points. For the 90% of the data set, 5% of data is used as validation set, i.e., 2405 data points.

### 6.3 Testing Set

The data set used to test the performance of the model. For the 80% training data set, 10% of the data is used for the testing set, i.e., 4810 data points. For the 90% of the data set, 5% is used for validation, i.e., 2405 data points. The error computed on the testing set is called testing error. Figure 6 describes the methodology used for the study.

The following steps show the implementation of the training algorithm:

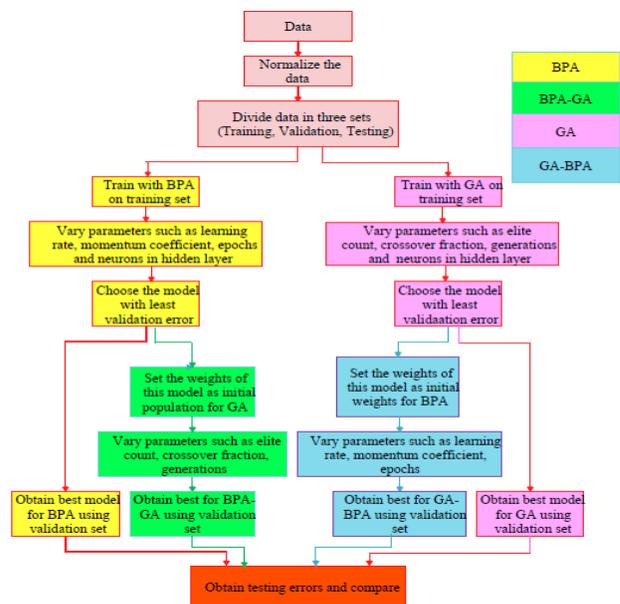


Figure 6. The methodology followed in this work.

After normalizing the data, BPA and GA are used to train the data sets

- Parameters such as (number of neurons, learning rate, momentum coefficient in BPA, number of neurons, elite count, cross over fraction in GA) affecting the performance of the architecture are analyzed.
- The algorithms are used to train the network with these parameters one at a time.
- The training error and the validation error are computed.
- The model with the least validation error is selected, and its parameters are fixed.
- The best results of BPA is selected is used to improve the accuracy of prediction by using GA (BPA-GA). [The weights of the best model of BPA are given as the initial population of the GA and the parameters for GA are varied]. Similarly, the best results of GA is selected and it is used to improve the accuracy of prediction by using BPA (GA-BPA). [The weights of the best model of GA are given as the initial weights of the BPA and the parameters for BPA are varied].
- The testing errors for the best models of BPA, GA, BPA-GA, GA-BPA are computed and compared.
- Best model for the prediction of wind power is selected.

### 6.4 Performance Index - Mean Squared Error (MSE)

The accuracy in prediction of wind power is evaluated by the performance index known as Mean Squared Error (MSE). The Mean Square Error is calculated by using (9). MSE is used as it gives a quadratic function in error. BPA, which uses steepest descent method converges for quadratic functions. Also derivative calculations are easy for the case of MSE, rather than RMSE. MSE is

scale-dependent and is widely used performance index in application of prediction problem.

$$MSE = \frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2 \tag{9}$$

Where,  $f_i$  is the prediction value and  $y_i$  is the actual data value,  $n$  is the total number of data sets. The predictions are more accurate when MSE is close to zero.

## 7. Results and Discussions

The two different learning algorithms of ANN, namely BPA and GA are used for prediction of wind power. In BPA, the learning rate, momentum coefficient, number of neurons in the hidden layer and epochs are varied. The total number of iterations are fixed to 2500. For the 80% training data, the best result among the 315 different combinations was found to be 5 neurons with 1500 epochs, learning rate as 0.5 and momentum coefficient as 0.1. This was chosen on the basis of least validation error of 0.00188 as shown in Table 1 and the corresponding testing error is 0.00223. The convergence graph for the best result for 80% training data set is shown in Figure 7.

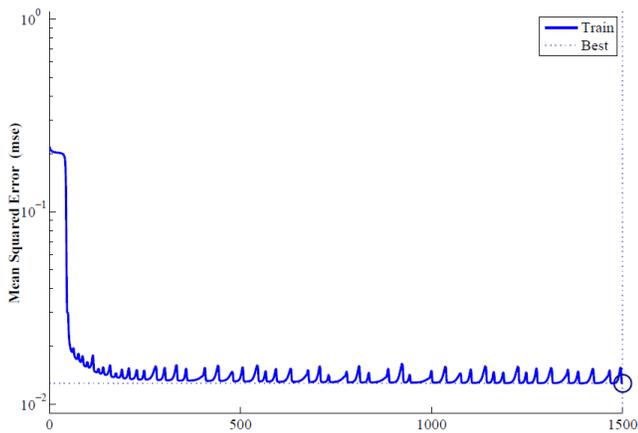
For the 90% training data, the best value of validation error is 0.00242 and the corresponding testing error is 0.00175. The testing error has decreased, by 15%, whereas the validation error has increased by 25%, for an increase in the training data set. The minimum value of validation error for both the training data set is obtained with 5 neurons. The best learning rate is obtained as 0.9 and the momentum coefficient is changed as 0.5, as shown in Table 1.

For GA, the parameters such as elite count and crossover fraction, along with the number of generations and the number of neurons in the hidden layer are varied. For the 80% of training set, the minimum validation error is obtained as 0.00127, and the corresponding testing error is 0.00166 as shown in Table 2. The best

**Table 1.** Best results for BPA with variation of training data sets

| % of Training Data | No. of Neurons | Epochs | Learning Rate | Momentum Coefficient | Training Error | Validation Error | Testing Error |
|--------------------|----------------|--------|---------------|----------------------|----------------|------------------|---------------|
| 80                 | 5              | 1500   | 0.5           | 0.1                  | 0.01298        | 0.00188          | 0.00223       |
| 90                 | 5              | 500    | 0.9           | 0.5                  | 0.01220        | 0.00242          | 0.00175       |

architecture obtained while using GA is 4 neurons in the hidden layer, 300 generations, the crossover fraction is 0.9 and elite count of 20%. For the 90% set, the minimum validation error is 0.00185 and corresponding testing error is 0.00132. The number of neurons in the hidden layer is 4 and generation is 400. The crossover fraction is same as for the 80%, but the elite count has decreased to 15%. There is an increase of 38% in validation error, and a decrease in testing error by 23.5%, as the training data is increased from 80% to 90% is shown in Table 2.



**Figure 7.** Best convergence for 80% training data.

**Table 2.** Best results for GA with variation of training data sets

| % of Training Data | No. of Neurons | Generations | Elite Count | Crossover Fraction | Training Error | Validation Error | Testing Error |
|--------------------|----------------|-------------|-------------|--------------------|----------------|------------------|---------------|
| 80                 | 4              | 300         | 0.20        | 0.9                | 0.01336        | 0.00127          | 0.00166       |
| 90                 | 4              | 400         | 0.15        | 0.9                | 0.01203        | 0.00185          | 0.00132       |

**Table 3.** Best results for BPA-GA with variation of training data sets

| % of Training Data | No. of Neurons | Generations | Elite Count | Crossover Fraction | Training Error | Validation Error | Testing Error |
|--------------------|----------------|-------------|-------------|--------------------|----------------|------------------|---------------|
| 80                 | 5              | 500         | 0.15        | 0.9                | 0.01290        | 0.00187          | 0.00223       |
| 90                 | 5              | 300         | 0.20        | 0.9                | 0.01177        | 0.00247          | 0.00156       |

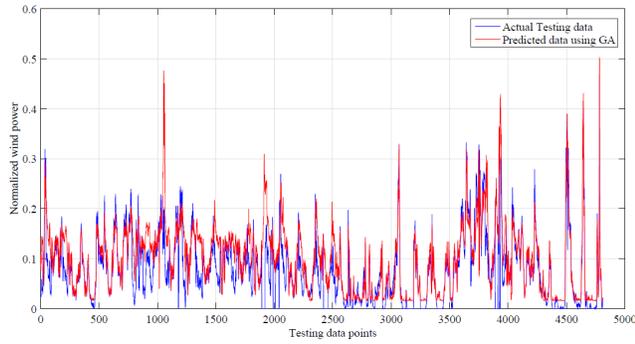
**Table 4.** Best results for GA-BPA with variation of training data sets

| % of Training Data | No. of Neurons | Epochs | Learning Rate | Momentum Coefficient | Training Error | Validation Error | Testing Error |
|--------------------|----------------|--------|---------------|----------------------|----------------|------------------|---------------|
| 80                 | 4              | 500    | 0.9           | 0.1                  | 0.01272        | 0.00228          | 0.00263       |
| 90                 | 4              | 500    | 0.9           | 0.5                  | 0.01169        | 0.00281          | 0.00228       |

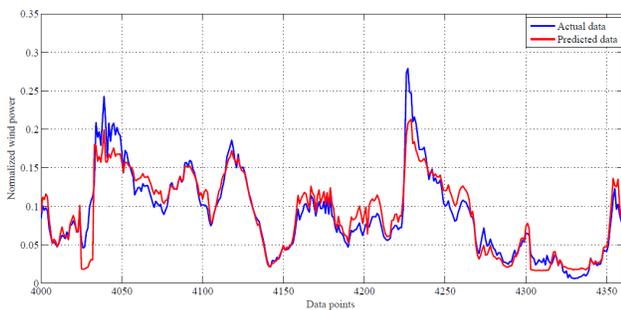
It can be inferred from the discussions that the GA has outperformed BPA in both the training data sets. Also a notable observation is that the validation error has increased when increasing the training data for both the algorithm but the testing error has decreased.

Also, in this study, the combination of the two models for prediction of wind power are proposed namely, BPA-GA and GA-BPA. For 80% of training set, BPA-GA method resulted in reduction of all three errors as shown in Table 3 only by a small margin. For 90% data set, the training and testing error decreased, meanwhile the validation error is increased. Overall, the hybrid BPA-GA improved the results for BPA but are not closer to the results obtained that of GA.

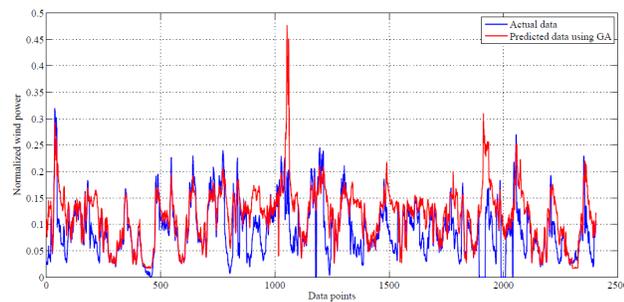
For GA-BPA, the training error in both the sets of training data has decreased, whereas the validation and testing error have increased greatly as shown in Table 4. This shows that this hybrid method is over fitting the data and fails to improve on the results of GA. This over fitting problem can be solved by using regularisation. The comparison of testing errors of all models used in this study are shown in Table 4. The best result of GA for 10% of testing data with the actual measured data is plotted and is shown in Figure 8 and for clarity it is presented again in Figure 9. Similarly the actual data sets and predicted



**Figure 8.** Actual data and predicted data using GA for 10% testing data.



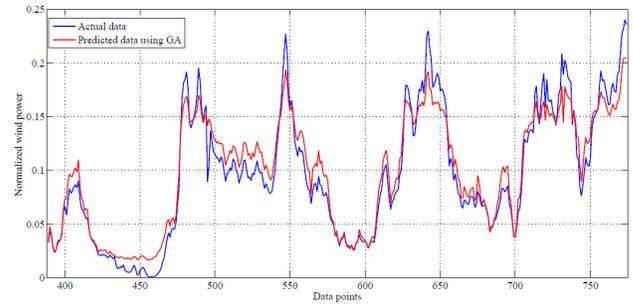
**Figure 9.** A section of Figure 8 of actual data and predicted data using GA.



**Figure 10.** Actual data and predicted data using GA for 5% testing data.

**Table 5.** Comparison of testing error for the four models for different training data sets

| % of Training Data | BPA     | GA      | BPA-GA  | GA-BPA  |
|--------------------|---------|---------|---------|---------|
| 80                 | 0.00223 | 0.00166 | 0.00223 | 0.00263 |
| 90                 | 0.00175 | 0.00132 | 0.00156 | 0.00228 |



**Figure 11.** A section of Figure 10 of actual data and predicted data using GA.

values using the best model i.e., using GA is plotted for 5% testing data is shown in Figure 10 and for clarity it is presented again in Figure 11.

## 8. Conclusions

In this study, four different models for prediction of wind power are presented. The two learning algorithms, such as back propagation algorithm and genetic algorithm, along with combination of these models are used for the prediction of wind power. Two training data sets are considered for the study to check the effectiveness of the algorithms. From the results it is concluded that;

- Genetic Algorithm performs better for both the training data sets than the other models.
- BPA can also be improved by the hybrid BPA-GA, but GA performs even better than the hybrid models.
- GA-BPA was not able to improve the results of GA as it over fits the data.
- Finally, it is recommended that the Genetic Algorithm as a learning algorithm in ANN is better suited for the prediction of wind power.

## 9. Acknowledgment

The authors would like to thank wind farm owner in Tamil Nadu for providing the real time wind data.

## 10. References

1. Global Wind Energy Council. Available from: <http://www.gwec.net/>

2. Mabel MC, Fernandez E. Analysis of wind power generation and prediction using ANN: A case study. *Renewable Energy*. 2008; 33(5):986–92.
3. Zhao X, Wang S, Li T. Review of evaluation criteria and main methods of wind power forecasting. *Energy Procedia*. 2011; 12:761–9.
4. Chang W-Y. A literature review of wind forecasting methods. *Journal of Power and Energy Engineering*. 2014; 2:161–8.
5. Lawan SM, Chai WAWZAWY, Baharun A, Masri T. Different models of wind speed prediction: A comprehensive review. *International Journal of Scientific and Engineering Research*. 2014 Jan 5:1760–68.
6. Foley AM, Leahy PG, Marvuglia A, McKeogh EJ. Current methods and advances in forecasting of wind power generation. *Renewable Energy*. 2012; 37(1):1–8.
7. Palomares-Salas J, De la Rosa J, Ramiro J, Melgar J, Aguera A, Moreno A. Arima vs. neural networks for wind speed forecasting. *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA '09)*; May 2009. p. 129–33.
8. Shi J, Qu X, Zeng S. Short-term wind power generation forecasting: Direct versus indirect arima-based approaches. *International Journal of Green Energy*. 2011; 8:100–12.
9. Khosravi A, Nahavandi S. An optimized mean variance estimation method for uncertainty quantification of wind power forecasts. *International Journal of Electrical Power and Energy Systems*. 2014; 61:446–54.
10. Skittides C, Fruh W-G. Wind forecasting using principal component analysis. *Renewable Energy*. 2014; 69:365–74.
11. Potter C, Negnevitsky M. Very short-term wind forecasting for Tasmanian power generation. *IEEE Transactions on Power Systems*. 2006 May; 21:965–72.
12. Lodge A, Yu X-H. Short term wind speed prediction using artificial neural networks. *4th IEEE International Conference on Information Science and Technology (ICIST)*; 2014 Apr. p. 539–42.
13. Flores P, Tapia A, Tapia G. Application of a control algorithm for wind speed prediction and active power generation. *Renewable Energy*. 2005; 30(4):523–36.
14. Sreelakshmi K, Ramakanth Kumar P. Neural networks for short prediction speed prediction. *World Academy of Science, Engineering and Technology*. 2008; 2(6):567–71.
15. Mohandes MA, Rehman S, Halawani TO. A neural networks approach for wind speed prediction. *Renewable Energy*. 1998; 13(3):345–54.
16. Cadenas E, Rivera W. Wind speed forecasting in the south coast of Oaxaca, Mexico. *Renewable Energy*. 2007; 32(12):2116–28.
17. Li M, Pan Y. Wind power prediction based on bpnn and lsa. *2012 Asia- Pacific Power and Energy Engineering Conference (APPEEC)*; 2012 Mar. p. 1–5.
18. Catalao J, Pousinho H, Mendes V. An artificial neural network approach for short-term wind power forecasting in Portugal. *15th International Conference on Intelligent System Applications to Power Systems (ISAP '09)*; Nov 2009. p. 1–5.
19. Fonte PM, Silva GX, Quadeado JC. Wind speed prediction using artificial neural networks. *Proceedings of the 6th WSEAS Int Conf on Neural Networks*; Lisbon, Portugal. 2005 Jun 16-18. p. 134–9.
20. Perez-Lleraa JFC, Fernandez-Baiznb MC, del Vallea VG. Local short-term prediction of wind speed: A neural network analysis. *International Environmental Modelling and Software Society (IEMSs)*; Lugano, Switzerland. 2002 Jun 24-27. p. 124–9.
21. Upadhyay KG, Choudhary AK, Tripathi MM. Short-term wind speed forecasting using feed forward back-propagation neural network. *International Journal of Engineering, Science and Technology*. 2011; 3(5):107–12.
22. Li G, Shi J, Zhou J. Bayesian adaptive combination of short-term wind speed forecasts from neural network models. *Renewable Energy*. 2011; 36(1):352–9.
23. Catalao J, Pousinho H, Mendes V. Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable Energy*. 2011; 36(4):1245–51.
24. Sheela KG, Deepa S. Neural network based hybrid computing model for wind speed prediction. *Neurocomputing*. 2013; 122:425–9.
25. Kim K-J, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*. 2000; 19(2):125–32.
26. Montana D, Davis L. Training feed forward neural networks using genetic algorithms. *Proc Eleventh Int Joint Conf on Artificial Intelligence*; San Mateo, CA: Morgan Kaufmann; 1989. p. 762–7.
27. Stastny J and Skorpil V. Genetic algorithm and neural network. *WSEAS Applied Informatics and Communications*; 2007. p. 347–51.
28. Suykens DJ, Vanderwalle J, Moor BD. Genetic weight optimisation of a feed forward neural network controller. *Artificial Neural Nets and Genetic Algorithms*. 1993:658–63.
29. Deshmukh M, Moorthy CB. Application of genetic algorithm to neural network model for estimation of wind power potential. *Journal of Engineering, Science and Management Education*. 2010; 2:42–8.
30. Diaz-Gomez PA, Hougen DF. Initial population for genetic algorithms: A metric approach. *Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods (GEM)*; Las Vegas, Nevada, USA. 2007 Jun 25-28. p. 43–9.
31. Roeva O, Fidanova S, Paprzycki M. Influence of the population size on the genetic algorithm performance in case of

- cultivation process modelling. 2013 Federated Conference on Computer Science and Information Systems (FedCSIS); 2013 Sep. p. 371–6.
32. Che Z-G, Chiang T-A, Che Z-H. Feed-forward neural networks training: A comparison between genetic algorithm and back-propagation learning algorithm. *International Journal of Innovative Computing, Information and Control*. 2011; 7(10):5839–51.
  33. Lin Xu R, Xu X, Zhe Hou X, Zhu B, You Chen M. A prediction model for wind farm power generation based on genetic-neural network. *JCIT*. Aug 2012; 7:11– 9.
  34. Kriesel D. A Brief introduction to neural networks. 2007.
  35. Bishop CM. *Neural networks for pattern recognition*. Clarendon Press Oxford; 1995.
  36. Chakraborty R. *Fundamentals of neural networks*. Jun 2010.
  37. Sathya R, Abraham A. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*. 2013; 2(2):34–8.
  38. Bishop CM. *Pattern recognition and machine learning*. Springer; 2006.
  39. Yan W, Liming Z. The effect of initial weight, learning rate and regularization on generalization performance and efficiency. 6th International Conference on Signal Processing. Aug 2002; 2:1191–4.
  40. Torii M, Hagan M. Stability of steepest descent with momentum for quadratic functions. *IEEE Transactions on Neural Networks*. 2002; 13(3):752–6.
  41. Mitchell M. *An introduction to Genetic Algorithms*. Cambridge, MA: MIT Press/Addison-Wesley; 1996.
  42. Gupta B, Dhingra S. Analysis of genetic algorithm for multiprocessor task scheduling problem. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2013; 3(7):339–44.
  43. Malik S, Wadhwa S. Preventing premature convergence in genetic algorithm using DGCA and elitist technique. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2014; 4(6):410–8.
  44. Pongcharoen P, Hicks C, Braiden P, Stewardson D. Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. *International Journal of Production Economics*. 2002; 78(3):311–22.
  45. Sola J, Sevilla J. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*; 1997 Jun. p. 1464–8.
  46. Hyndman RJ, Koehler AB. Another look at measures of forecast accuracy. *International Journal of Forecasting*. 2006; 22(4):679–88.