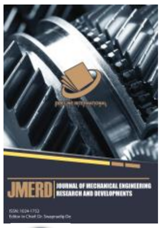
ZIBELINE INTERNATIONAL PUBLISHING

ISSN: 1024-1752 (Print)

CODEN: JERDFO

Journal of Mechanical Engineering Research & Developments (JMERD)

DOI: http://doi.org/10.26480/jmerd.01.2019.109.115



RESEARCH ARTICLE

CrossMark

EFFICIENT MODELLING AND SIMULATION OF WIND POWER USING ONLINE SEQUENTIAL LEARNING ALGORITHM FOR FEED FORWARD NETWORKS

Rashmi P Shetty[1,2*], A. Sathyabhama[1], Srinivasa Pai P[2]

[1]Department of Mechanical engineering, National Institute of Technology Karnataka, 575025, India
[2]Department of Mechanical engineering, NMAM Institute of Technology, Nitte, 574110, India
**Corresponding Author Email: iprashmi@nitte.edu.in*

ARTICLE DETAILS

Article History:

Received 1 January 2019
Accepted 22 February 2019
Available online 7 March 2019

ABSTRACT

In this paper, an online sequential learning algorithm known as online sequential extreme learning machine (OS ELM) is applied to simulate the power output of a wind turbine. The OS ELM is used both in 1-by-1 and chunk-by-chunk mode and the results are compared with batch learning algorithms, namely Back Propagation (BP) and Extreme Learning Machine (ELM) algorithm. Different activation functions such as Sigmoidal, Sin, Radial Basis Function (RBF) and Hardlim have been used in OS ELM to decide upon most optimal function. It has been found that OS ELM with fixed chunk size of 50-by-50 and sigmoidal activation function with training time of 0.080s, Root Mean Square Error (RMSE) of 1.96%, prediction accuracies on training and test data of 100% and 99.95 % respectively, is best suited for wind power modelling and simulation applications, where the data arrives in a sequential manner.

KEYWORDS

Extreme Learning Machine (ELM), Online Sequential ELM (OS ELM), Wind power modelling, Back Propagation (BP)

1. INTRODUCTION

Wind energy harvesting has gathered great momentum in recent decades, as it is found to be a feasible alternative to conventional energy sources and thus helps in reducing global warming problem. This promising alternate energy, though has significant social, environmental and economic impacts, suffers from uncertainty, due to intermittency and stochasticity involved in the wind, that restricts its integration with existing power system. It is a great challenge to develop an accurate and reliable prediction model, that will help to come out of this bottleneck for large scale penetration of wind power.

Numerous efforts of using parametric and nonparametric techniques are found in the literature in modelling and simulation of wind farm power output of a turbine. There are set of modelling efforts that tries to map the relationship between wind speed and the power obtained from the power curve of the turbine [1-5]. Due to the fact that the manufacturer's power curve is obtained under standard conditions and does not depict the performance of the turbine under actual working conditions, the second category by considering several other parameters affecting the wind turbine power, in addition to wind speed are developed. A researcher used input variables namely, wind direction and wind speed collected from two meteorological towers, to develop regression and artificial neural network (ANN) models [6]. On comparison it was found that ANN performed better than regression model. A previous researcher considered relative humidity, wind speed and generation hours as the input variables to the ANN model and observed that simulated results were close to actual [7,8]. Another researcher used BP and RBF neural networks for wind power prediction by considering three inputs namely, wind speed, direction and wind power as inputs and concluded that RBF perform better than BP [9]. An early scholar investigated most suitable input parameters to predict energy output of a wind farm having limited measured data [10]. Wind speed and power obtained in the previous time duration were found to be the potential input parameters. A previous study used wind direction and wind speed data to build a complex valued recurrent neural network model to predict wind turbine power and the model was observed to be highly accurate [11]. Another study considered wind speed, direction and temperature as input parameters for wind turbine power prediction using, neural network, k - nearest neighbor cluster center and fuzzy logic models [12]. It was proved that in developing models for predicting the wind turbine power, temperature and wind direction are the potential input parameters. A researcher-built models for prediction of wind farm power output by integrating evolutionary algorithms and data mining techniques [13]. The k-nearest model along with the principal component analysis (PCA) outperformed other techniques.

Most of these ANN models used BP, which is a most popular and widely used learning algorithm. Being iterative in nature, BP learning algorithm results in slow learning and gets stuck in local minima easily. To obtain better learning performance, it demands tuning of several simulation parameters. Recent study introduced a fast and non-iterative learning algorithm, known as Extreme Learning Machine (ELM) [14]. Learning using this algorithm takes place in one epoch, hence there is a drastic reduction in computational time especially for large data sets.

A group of researchers found that, wind power prediction models based on ELM are fast and accurate than BP [15]. Another researcher developed a RBF model to simulate the wind turbine power using ELM learning algorithm [16]. They used hybrid Particle Swarm Optimization (PSO) based Fuzzy C Means (PSO-FCM) clustering algorithm. The number of centers as well as the width of the RBF units were optimized by using PSO. The model developed was compact and efficient. Fast and improved generalization performance of ELM has opened the doors for ANN in online applications. Wind power prediction is one such area, where online sequential learning is very much suitable when compared to batch learning, since it avoids training the model, when a new set of observation is obtained. A previous scholar introduced online sequential ELM (OS ELM) with sigmoidal or RBF activation function in a unified framework, which can learn when the data arrives in 1-by-1 or chunk-by-chunk form [17]. OS ELM and many of its variants find diverse applications [18]. A researcher presented rolling ultra-short term of wind power prediction using an OS ELM algorithm by considering error interval evaluation and wind speed correction [19]. Three models were independently developed by using Numerical Weather Predicted (NWP) wind direction, air density and wind speed as the input parameters. It was observed that wind speed correction improves the accuracy and OS ELM algorithm has an advantage with regard to computation time compared to support vector machine (SVM)and BP.

As it is observed from the literature, most of the efforts in wind power modelling and prediction are based on batch learning either using BP or ELM, by using offline data. But the data acquired from the wind turbine is online in nature. Hence, OS ELM can be used effectively by considering newly arriving data for updating the network parameters online and make it suitable for online learning and avoid efforts in storing large amount of data. In one of the efforts, the authors have used OS ELM for wind power prediction by considering the NWP wind direction, air density and wind speed as the inputs. However, some of the significant input parameters affecting the wind turbine power, which includes rotor speed and blade pitch angle have not been considered, as they are related to wind turbine operation. Also, they have not considered other aspects of modelling using OS ELM with respect to the mode of data usage namely, 1-by-1 or chunk-by-chunk and use of different activation function, thus failing to fully exploit the modelling capabilities of the algorithm.

Thus, in the present work, an ANN model based on OS ELM algorithm has been used to simulate the power output of a wind turbine with carefully selected input parameters. The model capabilities of OS ELM have been studied in detail with respect to different modes of learning namely 1-by-1, chunk-by-chunk, and the use of different activation functions namely, Sigmoidal, Sin, Radial Basis Function (RBF) and Hardlim. The performance is then compared with models based on BP and ELM batch learning algorithms to propose a most suitable ANN model for wind power prediction application. The model that is found to be most accurate and suitable has been validated. The data stored in a supervisory control and data acquisition (SCADA) system of a wind farm present in central dry zone of Karnataka state, India has been used.

2. FEEDFORWARD NEURAL NETWORK (FNN)

FNN, a simplest type of ANN, has the ability to accurately map complex nonlinear relationships. Hence it finds extensive application in many fields [20]. The FNN consists of three layers where, the input layer is connected to the external world to receive the information, hidden layer uses some activation function to process the data and the output layer provides the output. The synaptic weights between input, hidden and output layer are modified during the learning process. The FNN consists of connections only in the forward direction from input through hidden to the output layer. The information does not flow in backward direction [21]. Various activation functions namely Sigmoidal, Sin, RBF and Hardlim as given in equations (12), (13), (14) and (15) can be used. Out of these, the Sigmoidal and RBF are most widely used.

$$V_j^{\mu} = \frac{1}{1+e^{\sum w_{ji}x_i+b}} \tag{12}$$

$$V_j^{\mu} = \sin(\sum w_{ji}x_i + b) \tag{13}$$

$$V_j^{\mu} = e^{-\frac{1}{2\sigma^2}\left|\left|x_j-x_i\right|\right|^2} \tag{14}$$

$$V_j^{\mu} = \begin{cases} 1, & if\ input\ reaches\ threshold \\ 0, & otherwise \end{cases} \tag{15}$$

where, x_i is the input data, x_j, σ are center, width of the RBF unit, $i = 1,2,...p$, $\mu = 1,2,..n$, j = 1,2,3...s and $k = 1,2,...r$ are the input features, numbers of patterns, number of hidden layer neurons and output features respectively, b is the bias and w_{ji} is the weights between hidden and input layer.

The output of the network is then calculated using equation (16)

$$O_j^{\mu} = \frac{1}{1+e^{\sum w_{kj}V_j+b}} \tag{16}$$

where, w_{jk} is the weights between hidden and output layer.

One of the properties of the FNN is its capacity to learn from the environment. In case of batch learning, all the training samples are fed at once to the network. Two commonly used batch learning algorithms are

i) Backpropagation algorithm
ii) Extreme learning machine algorithm

2.1 Backpropagation algorithm

In BP learning, the weights are updated in an iterative manner based on the error signal that is the difference between the actual and the network output. The weights are updated as given in equation (17).

$$w_{kj}(n + 1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{17}$$

where $\Delta w_{kj}(n)$, the adjustment applied to the synaptic weights, is calculated using equations (18) and (19)

$$\delta_k(n) = \big(O_k(n) - y_k(n)\big)O_k(n)(1 - O_k(n)) \tag{18}$$

$$\Delta w_{kj}(n) = \delta_k(n)V_j(n)\eta\alpha \tag{19}$$

where, α is the momentum parameter, η is the learning rate, $V_j(n)$ is the hidden layer output and $y_k(n)$ is the target output. This algorithm is generally slow and convergence early to local minima. It needs tuning of many network parameters namely weights and bias iteratively.

2.2 Extreme learning machine algorithm

The ELM algorithm introduced by a researcher draws the attention of researchers by overcoming the limitations of BP algorithm. This is a non-iterative learning algorithm, where the learning takes place in only one epoch using the Moorse- Penrose generalized inverse operation. Due to this reason, it is extremely faster in comparison to BP. The algorithm overcomes the problem of getting stuck in local minima and overtraining. Hence results in good generalization performance.

2.2.1 Description

Let there be N arbitrary distinct samples (x_i, t_i), here x_i is input vector and t_i is target vector and Q hidden nodes with activation function g(x). The network output O_j can be mathematically expressed as given in equation (20).

$$O_j = \sum_{i=1}^{Q} \beta_i g_i(w_i x_j + b_i) \text{ , j=1,..., N.} \tag{20}$$

where w_i is the weights between input and hidden neurons, b_i is the bias of i[th] hidden neuron and β_i is the output weight matrix ie. weight between the hidden and output layer neurons. Equation (20) can be expressed as equation (21).

$$H\beta=T \tag{21}$$

where H and T are hidden layer output and target matrix respectively. H, β and T are given in equations (22), (23) and (24) respectively.

$$H(w_1,...w_N,b_1,....b_N,x_1,.....x_N)= \begin{bmatrix} g(w_1x_1+b_1) & \ldots & g(w_Qx_1+b_Q) \\ . & & . \\ . & \ldots & . \\ . & & . \\ g(w_1x_N+b_1) & \ldots & g(w_Qx_N+b_Q) \end{bmatrix}_{NXQ} \tag{22}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \cdot \\ \cdot \\ \cdot \\ \beta_Q^T \end{bmatrix}_{QXm} \quad (23) \qquad T = \begin{bmatrix} t_1^T \\ \cdot \\ \cdot \\ \cdot \\ t_N^T \end{bmatrix}_{NXm} \quad (24)$$

The output weight matrix is found by equation (25).

$$\beta = H^{\dagger}T \tag{25}$$

where $H^{\dagger}$ is the Moorse- Penrose generalized inverse of matrix H, which can be determined using equation (26).

$$H^{\dagger} = \left(H^T H \ \right)^{-1} H^T \tag{26}$$

3. ONLINE SEQUENTIAL EXTREME LEARNING MACHINE

However, both BP and ELM are batch learning algorithms, demanding complete training data to develop the model. But in many of the industrial applications, the training data arrives in a sequential manner appealing for online sequential learning rather than batch learning. The OS ELM algorithm proposed by a researcher can learn from online sequential training data arriving 1-by-1 or chunk-by-chunk. Both additive and RBF nodes, which can use any bounded nonconstant piecewise continuous activation functions and any integrable piecewise continuous activation functions respectively, can be handled by this versatile learning algorithm. It discards the training observation(s) immediately as the learning action for that observation(s) is complete. The OS ELM algorithm is discussed below.

It works in two stages namely an initialization stage and a sequential learning stage.

Step 1. Initialization stage:

a) Initially feed a small block of data N0, which is not smaller than number of hidden neurons, from the given set of training data.
b) Randomly allocate input weights, bias, (for sigmoidal activation function) or center, widths (for RBF activation function)
c) Determine the hidden layer outputs
d) Calculate the initial output weight matrix using equation (27)

$$\beta_{(0)} = P_0 H_0^T T_0 \tag{27}$$

where P_0 and T_0 are given in equations (28) and (29) respectively. H_0, T_0 are hidden layer output and target matrix respectively for initial block of data N0.

$$P_0 = (H_0^T H_0)^{-1} \tag{28}$$

$$T_0 = \begin{bmatrix} t_1^T \\ \cdot \\ \cdot \\ \cdot \\ t_0^T \end{bmatrix} \tag{29}$$

Step 2. Sequential learning stage:

Present the (k+1)th chunk of new data

a) Find the outputs of hidden layer using equation (30)
b)

$$H_{k+1} = \begin{bmatrix} g(w_1 x_{\left(\sum_{j=0}^{k} N_j\right)+1} + b_1) & \dots & g(w_Q x_{\left(\sum_{j=0}^{k} N_j\right)+1} + b_Q) \\ \cdot & & \cdot \\ \cdot & \dots & \cdot \\ \cdot & & \cdot \\ g(w_1 x_{\sum_{j=0}^{k+1} N_j} + b_1) & \dots & g(w_Q x_{\sum_{j=0}^{k+1} N_j} + b_Q) \end{bmatrix}_{NXQ} \tag{30}$$

b) Determine the output weight matrix $\beta^{(k+1)}$ using equation (31) and (32).
c)

$$P_{K+1} = P_K - P_K H_{K+1}^T (I + H_{K+1} P_K H_{K+1}^T)^{-1} H_{K+1} P_K \tag{31}$$

$$\beta^{K+1} = \beta^{(K)} + P_{K+1} H_{K+1}^T (T_{K+1} - H_{K+1}\beta^{(K)}) \tag{32}$$

4. WIND POWER MODELLING

In a wind turbine, wind interacts with blades of the turbine, thus rotating the rotor, which results in power generation. Wind turbine power is mainly a function of wind speed. But it is also true that there are a few other parameters, which have significant effect on power generation. The theoretical power from a wind turbine is expressed using Equation (33)

$$\mathrm{P} = \frac{1}{2}\rho\pi \mathrm{R}^2 \mathrm{C_p}(\lambda, \beta)\mathrm{v}^3 \tag{33}$$

where,

P is the power captured by the rotor of a wind turbine in kW, ρ is the density of air in kg/m^3, R is the rotor radius in m, C_p is the power coefficient, β is the blade pitch angle in degree, λ is the tip speed ratio, v is the wind speed in m/s

C_p is a function of tip speed ratio λ and blade pitch angle β. The tip speed ratio is defined as the ratio of the tangential velocity of the blade tips and the effective wind speed. λ can be determined using Equation (34) [22].

$$\lambda = \frac{\mathrm{R}\Omega_\mathrm{r}}{\mathrm{V_e}} \tag{34}$$

where R is the rotor radius in meters, Ω_r is the rotor speed in radians/second and V_e is the effective wind speed perpendicular to the rotor plane in m/s.

The wind direction is neglected in equation (33) due to the assumption that wind blows orthogonally to the rotor. But it is not the case, because wind blows from different directions in practice. Hence, it is an important variable to be considered. Accordingly, wind speed, rotor speed, blade pitch angle, direction and density are the parameters directly influencing the wind power generation. Out of these parameters blade pitch angle is one of the important controllable parameters.

5. DATA DESCRIPTION, PRE-PROCESSING AND MODEL DEVELOPMENT

The advancement in data acquisition systems have provided golden opportunities for the researchers. SCADA system is one of the vital components of modern wind farms as it is a means of acquiring online data about wind power generation. The data for the present study is collected from SCADA of a large wind farm of Karnataka state, India. This region is ideally suited for harvesting wind energy. A ten-minute resolution data from a 1500 kW, three bladed, pitch regulated upwind horizontal axis wind turbine, collected for a duration of six months during June - December 2013, has been used. With regard to enhancing the accuracy, the missing and erroneous data as a result of failure of sensors and several subsystems have been removed from the data set. After the data is averaged to 1 hour, it has been normalized between 0 and 1, so that each data contributes equally towards the learning process of ANN, by using equation (35).

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{35}$$

where, X ', X, X_{max} and X_{min} are normalized data, actual data, maximum and minimum values in the data set respectively.
The input variables considered are density, wind speed, rotor speed, wind direction and blade pitch angle. The output is wind turbine power. The general architecture of the neural network model is as shown in the Fig 1. The simulations have been executed in MATLAB R2014a environment using an Intel i-3, 2.2 GHZ CPU [23]. For each case, the results are averaged over 10 trials. Mean Square Error (MSE) and Root Mean Square Error (RMSE) given in the equations (36) and (37) respectively have been used in this work. RMSE has been used as a performance metric to assess different models. X_i and $\widetilde{X}_i$ are the ith element of actual and predicted values respectively and n is the total number of data. For assessing the prediction accuracy of developed models, a MSE value of 0.01 kW^2 has been considered.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \widetilde{X}_i\right)^2 \tag{36}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X_i})^2}{n}} \quad (37)$$

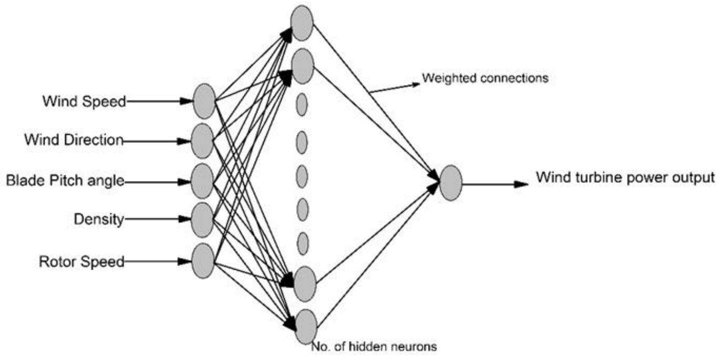


Figure 1: General architecture of the ANN model

6. RESULTS AND DISCUSSION

This work aims to apply OS ELM to model and simulate wind power. The results have been compared with ANN models based on BP and ELM batch learning algorithms.

6.1 OS ELM Model

Determining the optimal value of learning parameters and network architecture is an important step in ANN model development. The optimal number of hidden neurons is the only parameter to be tuned in OS ELM model. In the present work, it has been determined based on trial and error for achieving minimum RMSE (%). The variation of RMSE (%) with number of hidden neurons is shown in Fig 2. Decrease in RMSE (%) is observed with increase in number of hidden neurons. The lowest RMSE (%) is obtained for 30 neurons, and it increased thereafter. This behavior was observed for both Sigmoid and RBF activation functions. Accordingly, the optimum number of hidden neurons has been set to 30.

The performance of OS ELM for three different learning modes namely 1-by-1, use of fixed chunk size and randomly varying chunk size has been compared, considering two widely used activation functions namely Sigmoid and RBF. The corresponding results have been summarized in Table 1. A fixed chunk size of 50 and random chunk size in the range of [10,100] has been used. From the table it is found that the time taken for training using 1-by-1 is longest and it reduced for chunk- by-chunk case for both Sigmoid and RBF activation functions. For instance, for the Sigmoid case, the 1-by-1 learning mode takes 0.159 s but it is decreased to 0.080 s for 50-by-50 case. It can be noted that time taken for RBF function is much higher in comparison to Sigmoid activation function. This is because Sigmoid function satisfies a property given in (38), between its derivative f' and itself. Thus, it is computationally easy to perform when compared to RBF which uses Gaussian function. Higher RMSE (%) is observed for RBF function over Sigmoidal function for all learning modes. However, there is no noticeable difference in the prediction accuracies for training and test data using the two activation functions, because the calculation is based on a fixed MSE of 0.01 kW^2. Hence, it can be noted that use of chunk size 50 gives optimal performance for the considered wind data.

$$f' = f(1-f) \quad (38)$$

Table 1: Performance comparison of different learning modes of OS ELM

Activation functions	Algorithm	Learning mode	Time (Seconds)	RMSE (%)	Accuracy on training data (%)	Accuracy on test data (%)
Sigmoid	OS ELM	1-by-1	0.159	1.98	100	99.93
		50-by-50	0.080	1.96	100	99.95
		[10,100]	0.083	1.97	100	99.93
RBF	OS ELM	1-by-1	0.462	2.01	100	99.89
		50-by-50	0.097	2.04	100	99.86
		[10,100]	0.092	2.01	100	99.93

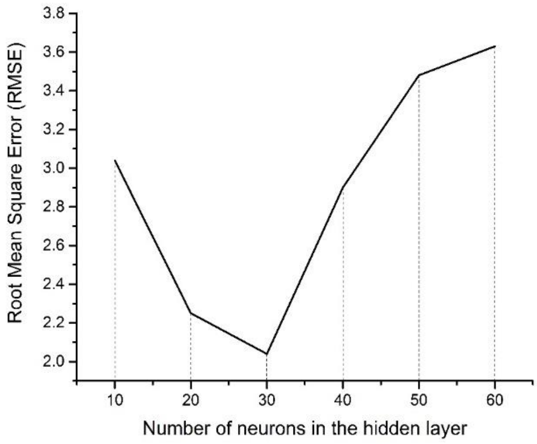


Figure 2: Variation of RMSE with number of hidden neurons

6.2 Comparison of use of different activation functions in OS ELM

To further understand the modeling behavior of OS ELM, four different activation functions have been investigated namely Sin and Hardlim in addition to widely used RBF and Sigmoid functions. The results are tabulated in Table 2, for fixed chunk size of 50-by-50. Hardlim activation function resulted in poor prediction accuracy of 64.83% and 63.86% for training and test data respectively, when compared to RBF, sigmoid and sin. RMSE is observed to be 21.69 % which is too high, compared to other three. Sigmoid activation function gives the best possible performance of 100% and 99.95% accuracy on training and test data respectively, with least RMSE. The computational time for use of RBF function is higher compared to other activation functions.

Table 2: Performance of OS ELM with different activation functions

	RBF	Sigmoid	Sin	Hardlim
Accuracy on training data (%)	100	100	100	64.83
Accuracy on test data (%)	99.86	99.95	99.91	63.86
RMSE (%)	2.04	1.96	2.01	21.69
Time (s)	0.097	0.080	0.076	0.083

6.3 Comparison of performance of OS ELM with other models

Performance of different learning modes of OS ELM namely1-by-1, fixed chunk size of 50-by-50 and random chunk size between [10,100], has been compared with models using ELM and BP batch learning algorithms. The results are presented in Table 3. Sigmoid and RBF activation functions have been considered for the performance comparison as they are the most widely used activation functions. The models using batch learning algorithms namely ELM and BP with Sigmoid and RBF activation function have been simulated using customized MATLAB codes. Hence the time taken by these algorithms are not comparable with that of OS ELM, which makes use of functions available in MATLAB ANN tool box. It has been observed by a researcher that the training time taken by batch learning is shortest and for sequential learning in 1-by-1 learning mode OS ELM is the longest. It is preferable to compare the two batch learning algorithms namely ELM and BP, which are widely used in ANN modelling applications. The learning in BP is iterative in nature and suffers from problems of easy convergence to local minima and hence results in slow training and poor generalization performance. ELM overcomes the drawbacks of BP algorithm and results in quick learning in 0.458 s and results in comparatively high prediction accuracies of 99.92 % and 99.54% for training and test data respectively in comparison to 21.54 s training time, 97.10% and 96.17% accuracy for training and test data respectively with the use of BP algorithm, when sigmoid activation function is used. Similar trend has been observed with use of RBF activation function.

Since BP is a gradient descent learning algorithm and suffers from local minima, it results in large number of RBF centers in the hidden layer [24]. A drastic reduction in number of RBF centers can be observed, when ELM algorithm is used. But there is no much difference in the number of hidden neurons, when sigmoid function is used during training using ELM algorithm [25].

From the literature it can be observed that, the prediction accuracies and size of RBF neural network can be improved to a great extent by proper selection of centers using suitable center selection strategy, there by resulting in a more robust and compact model. It is thus concluded from the table that OS ELM algorithm with fixed chunk size of 50 is superior, when compared to all other models with regard to prediction accuracies as well as the training time. Thus, it is the most suitable model, being fast and accurate for real world wind power modelling and simulation applications with data arriving in a sequential manner.

Table 3: Performance comparison of different ANN models

Activation functions	Algorithm	Learning mode	Time (Seconds)	RMSE (%)	Accuracy on training data (%)	Accuracy on test data (%)	Number of hidden neurons
Sigmoid	OS-ELM	1-by-1	0.159	1.98	100	99.93	30
		50-by-50	0.080	1.96	100	99.95	30
		[10,100]	0.083	1.97	100	99.93	30
	ELM	Batch	0.458	2.61	99.92	99.54	25

	BP	Batch	21.54	5.93	97.10	96.17	25
RBF	OS-ELM	1-by-1	0.462	2.01	100	99.89	30
		50-by-50	0.097	2.04	100	99.86	30
		[10,100]	0.092	2.01	100	99.93	30
	ELM	Batch	0.631	1.96	100	99.77	75
	BP	Batch	47.23	5.23	97.06	96.39	1125

6.3.1 Validation

The OS ELM model with sigmoidal activation function with fixed chunk size of 50 has been validated by taking 5% of the entire data set which has not been used in training as well as test data set. This data has been collected during May 2017, from an identical turbine in the same wind farm. The model is proved to be efficient with 97.68% prediction accuracy and RMSE of 5.12%. The corresponding results are shown in Fig 3 for 50 number of randomly selected data. There is good agreement between predicted and actual data.

There are little efforts in modelling and simulation of the wind turbine power based on OS ELM. To prove the reliability of the model developed, the results of the present work has been compared with one of the work available in the literature, where OS ELM has been used for wind power prediction. A researcher presented rolling ultra-short term of wind power prediction based on an OS ELM algorithm by considering NWP wind speed correction and error interval evaluation. The different modes of learning of the model, 1-by-1 or chunk-by-chunk is unaddressed in this work. The RMSE for the optimal OS ELM model and the data set combination ie. OS ELM model after wind speed correction and summer season data is found to be 5.08%, which is much higher in comparison to RMSE of 1.96 % of the OS ELM model developed in the present study.

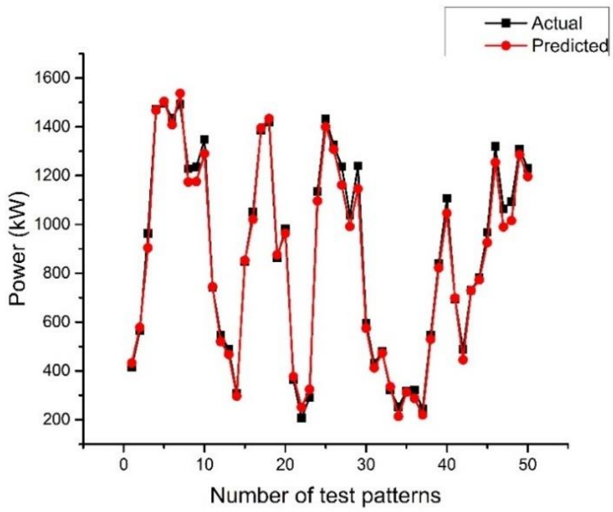


Figure 3: Comparison of simulated power with actual (Validation data)

7. CONCLUSIONS

A fast and accurate ANN model has been developed to simulate the power output of a horizontal axis wind turbine based on online sequential algorithm namely, OS ELM. Different activation functions namely Sigmoid, Sin, RBF and Hardlim have been used and the performance has been compared. In OS ELM, different modes of learning namely 1-by-1and chunk-by-chunk have been investigated. The performance of the OS ELM model has been compared with models based on ELM and BP batch learning algorithms, by considering sigmoidal and RBF, the most widely used activation functions.

From the study, the following conclusions can be drawn:

- In OS ELM, the training time taken for 1-by-1 mode of learning is high in comparison to chunk-by-chunk.
- OS ELM model with RBF activation function takes more training time compared to Sigmoid, Sin and Hardlim activation functions.
- Sigmoid, Sin and RBF activation functions showed good generalization performance when compared to Hardlim.
- Batch learning using ELM learning algorithm is better than BP, as it overcomes many of its drawbacks like slow learning, getting stuck in local minima and hence reduced number of hidden neurons, particularly when RBF activation function is used.
- OS ELM using 50-by-50 chunk size with Sigmoid activation function gives the optimal performance with 99.95% accuracy on test and 97.68% on validation data with least RMSE and training time.
- Though the batch learning using ELM with RBF activation function can result in a more robust neural network model with the proper use of center selection strategy [16], an ANN model based on OS ELM is more suitable for simulation and modelling of wind power data, where it arrives in a sequential manner, through a SCADA, thus reducing the effort and cost in storing large amount of data.

Thus, the study provides a detailed understanding of all aspects of wind power modelling and simulation with respect to different activation functions, modes of learning and learning algorithms, thereby helps in identifying a reliable and efficient ANN model for wind power prediction.

REFERENCES

[1] Lydia, M., Suresh Kumar, S., Immanuel Selvakumar, A., Edwin Prem Kumar, G. 2014. A comprehensive review on wind turbine power curve modeling techniques, Renewable Sustainable Energy Review, 30, 452–460.

[2] Burton, T., Jenkins, N., Sharpe, D., Bossanyi, E. 2011. Wind Energy Handbook.

[3] Shokrzadeh, S., Jozani, M.J., Bibeau, E. 2014. Wind Turbine Power Curve Modeling Using Advanced Parametric and Nonparametric Methods", IEEE Transactions on Sustainable Energy, 5, 1262–1269.

[4] Marvuglia, A., Messineo, A. 2012. Monitoring of wind farms' power curves using machine learning techniques, Applied Energy, 98, 574–583.

[5] Üstüntaş T., Şahin, A. D. 2008. Wind turbine power curve estimation based on cluster center fuzzy logic modelling, Journal of Wind Engineering and Industrial Aerodynamics, 96, 611-620.

[6] Shuhui, L., Donald, C., Wunsch, Edgar O'Hair, W., Giesselmann, M.G. 2001. Comparative Analysis of Regression and Artificial Neural Network

Models for Wind Turbine Power Curve Estimation, Journal of solar energy engineering, 123, 327-332.

[7] Mabel, M.C., Fernandez, E. 2008. Analysis of wind power generation and prediction using ANN: A case study, Renewable Energy, 33, 986–992.
[8] Mabel, M.C., Fernandez, E. 2009. Estimation of Energy Yield from Wind Farms Using Artificial Neural Networks, IEEE Trans. Energy Convers, 24, 459–464.

[9] Han, S., Yang, Y., Liu, Y. 2007. The comparison of BP network and RBF network in wind power prediction application, In Bio-Inspired Computing: Theories and Applications, 173-176.

[10] Tu, Y.L., Chang, T.J., Chen, C.L., Chang, Y.J. 2012. Estimation of monthly wind power outputs of WECS with limited record period using artificial neural networks, Energy Convers. Manage., 9, 114–121.

[11] Liu, Z., Gao, W., Wan, Y.H., Muljadi, E. 2012. Wind power plant prediction by using neural networks, in 2012 IEEE Energy Conversion Congress and Exposition (ECCE), 3154-3160.

[12] Schlechtingen, M., Santos, I.F., Achiche, S. 2013. Using Data-Mining Approaches for Wind Turbine Power Curve Monitoring: A Comparative Study, IEEE Transactions on Sustainable Energy, 4, 671–679.

[13] Kusiak, A., Zheng, H., Song, Z. 2009. Models for monitoring wind farm power, Renewable Energy, 34, 583–590.

[14] Huang, G.B., Zhu, Q.Y., Siew, C.K. 2014. Extreme learning machine: a new learning scheme of feedforward neural networks, in 2004 IEEE International Joint Conference on Neural Networks, 2, 985-990.

[15] Zheng-zhong, Z., Yi-min, J., Wen-hui, Z., Tian, X. 2014. Prediction of Short-Term Power Output of Wind Farms Based on Extreme Learning Machine, in Unifying Electrical Engineering and Electronics Engineering, Springer New York, 1029–1035.

[16] Rashmi P. Shetty, A. Sathyabhama, Srinivasa Pai P., Adarsh Rai, A. 2016. Optimized Radial Basis Function Neural Network model for wind power prediction, in 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), 1-6,

[17] Liang, N.Y., Huang, G.B., Saratchandran, P., Sundararajan, N. 2006. A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw., 17, 1411–1423.

[18] Wang, X., Han, M. 2014. Online sequential extreme learning machine with kernels for nonstationary time series prediction, Neurocomputing., 145, 90–97.

[19] Guo, L., Wang, C., Gao, P., Wang, Y., Zhong, Y., Huang, M. 2014. An online short-term wind power prediction considering wind speed correction and error interval evaluation, in 2014 International Conference on Information Science, Electronics and Electrical Engineering, 1, 28-32.

[20] Entchev, E., Yang, L., Ghorab, M., Rosato, A., Sibilio, S. 2018. Energy, economic and environmental performance simulation of a hybrid renewable microgeneration system with neural network predictive control, Alexandria Engineering Journal, 57, 455–473.

[21] Haykin, S. 1994. Neural networks: a comprehensive foundation. Prentice Hall PTR.

[22] Ramachandra, T.V., Shruthi, B.V. 2005. Wind energy potential mapping in Karnataka, India, using GIS, Energy Convers. Manage., 46, 1561–1578.

[23] Manwell, J.F., McGowan, J.G., Rogers, A.L. 2010. Wind Energy Explained: Theory, Design and Application. John Wiley & Sons.

[24] MATLAB R2014a @R, www.matheworks.com

[25] Pai, P.S. 2004. Acoustic emission-based tool wear monitoring using some improved neural network methodologies, Ph.D. dissertation, Dept. Mech. Eng., Mysore, India.