ISSN: 2251-9246
EISSN: 2345-622
pp. 101:107

Hourly Wind Speed Prediction using ARMA Model and Artificial Neural Networks

Farzaneh Tatari *, Majid Mazouchi

Electrical Engineering Department, Semnan University, Semnan, Iran, ftatari@semnan.ac.ir
Electrical Engineering Department, Ferdowsi University, Mashhad, Iran. mazouchi.majid@mail.um.ac.ir

Abstract

In this paper, a comparison study is presented on artificial intelligence and time series models in 1-hour-ahead wind speed forecasting. Three types of typical neural networks, namely adaptive linear element, multilayer perceptrons, and radial basis function, and ARMA time series model are investigated. The wind speed data used are the hourly mean wind speed data collected at Binalood site in Iran. Simulation results indicate the ability of the proposed methods in 1-hour-ahead wind speed forecasting in Binalood of Iran.

Keywords: Adaptive linear element (ADALINE); Artificial neural network (ANN); Auto-regressive moving average (ARMA); Multilayer perceptron (MLP); Radial basis function (RBF); Wind speed forecasting.
Article history: Received 19-DEC-2017; Revised 25-DEC-2017; Accepted 10-JAN-2018.

1. Introduction

Renewable wind energy promises a fast growing green and clean energy source. Wind speed stochastic nature is a significant issue for converting wind-energy to a reliable source of electric power. Wind power uncertainty and intermittency is a challenge in integrating wind power generation in to the electric grid. Wind speed randomness results in a random output power of the wind farm, therefore accurate wind speed forecasting is essential for power generation efficiency and mitigation of undesired results in wind energy conversion system.

Accurate wind forecasting deals with a lot of problems such as competitive market designs, real-time grid operations, standards of interconnection, quality of power, capacity of transmission system, stability and reliability of power system and optimal reductions in greenhouse gas emissions [1,2,3]. In competitive electricity markets, there are several important reasons to have an accurate wind forecast. Firstly, market prices are offered based on energy imbalance charges. Secondly, a correct forecast can help to develop well-functioning hour-ahead or day-ahead markets [3].

Time-scale classification of wind forecasting methods is vague. However, as mentioned in [8], wind forecasting can be categorized into 4 horizons. These categories include very short-term forecasting (few seconds to 30 minutes), short-term forecasting (30 minutes to 6 hours), medium-term forecasting (6 hours to 1 day) and long-term forecasting (1 day to 1 week).

In very short-term forecasting for periods in the range of a few seconds up to several minutes, the objective is the control of wind turbine. Short-term forecasting of wind speed is important for improving the efficiency of a wind power generation system [5] as well as for the integration of wind energy in to the power system [6,7]. Medium-term wind speed forecasts in the range of hours target the problem of scheduling in a power system, where forecasts in the range of days are related with maintenance and resource planning [9]. Long-term wind speed prediction is vital for the sitting and sizing of wind power applications [4]. Therefore improving wind power prediction has significant economic and technical advantages.

A suitable wind speed forecasting approach is determined based on the required time scale, its application and the available wind speed data. Several approaches have been developed for wind speed forecasting including physical methods, time series methods, hybrid models, artificial intelligence, etc. [10, 11, and 20]. Time series and artificial intelligence forecasting methods are based on training the predictor with measured data and calculating the difference between the predicted and the actual wind speeds in immediate past to tune predictor parameters.

Time series models make forecasts by finding the relationship of the observed wind speed time series [12, 13]. There are several prediction time-series models containing auto-regressive moving average (ARMA), auto regressive integrated moving average (ARIMA), seasonal- and fractional-ARIMA and ARMA with exogenous input (ARMAX or ARX) [19]. In time-series based approaches, ARMA models are widespread employed for forecasting. In [21], the authors have employed moving average (MA), weighted moving average (WMA), autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) methods for forecasting 10-min and 2-h ahead wind speed forecasting. The work of [13] used the ARMA model to predict the hourly average wind speed up to 10 hours in advance across the entire territory of the Regional Community of Navarre (Spain).

Artificial intelligence methods forecast the wind speed by learning the wind model based on the collection of input/output data pairs. Artificial neural networks (ANNs) are trained to learn the relationship between the input and output data. Neural networks (NNs) structure is composed of several layers containing an input layer, hidden layers and an output layer. The input layer receives the training data for learning and hidden layers and output layer provide the forecasted outputs. Several NN models exist named as feed-forward neural networks (FNNs), multi-layer perceptrons (MLP), recurrent neural networks (RNNs), radial basis function (RBF) NNs, adaptive linear element (ADALINE) networks, etc.

ANNs of multi-layer perceptrons [14], radial basis function [15] and recurrent neural networks [16] are widely employed for wind speed forecasting. The work of [22] employed different multilayer feed forward backpropagation networks to predict the short-term wind speed in Eskisehir region of Turkey. The authors in [23] studied the performance of ARIMA model and ANNs to forecast the hourly wind speed one to four hours ahead in a wind park in Hubei province of China. The work of [24] compared ARIMA, ANNs and polynomial curve fitting models for short term wind speed prediction.

The main contribution of the paper is the comparison study of 4 prediction methods on 1-hour-ahead wind speed prediction of Binalood. These four methods include three artificial neural networks, multilayer perceptron, radial basis function and adaptive linear element and one time series model called auto-regressive moving average. All the intended models are developed using hourly wind speed data of Binalood site, located in Khorasan-Razavi province of Iran, from January 1, 2010 to December 31, 2010.

2. Time Series Forecasting Model

In this paper ARMA time series model is employed for hourly wind speed prediction in Binalood. ARMA model structure and training process is explained as follows.

ARMA model has two components: auto-regressive and moving average. The ARMA model is characterized as

$$
\begin{aligned}
& y(\mathrm{t})+a_1 y(\mathrm{t}-1)+\ldots+a_{n_a} y(\mathrm{t}-n_a)= \\
& e(\mathrm{t})+c_1 e(\mathrm{t}-1)+\ldots+c_{n_c} e(\mathrm{t}-n_c), \\
& A(q) y(\mathrm{t})=C(q) e(\mathrm{t}) , \\
& A(q)=1+a_1 q^{-1}+\ldots+a_{n_a} q^{-n_a} , \\
& C(q)=1+c_1 q^{-1}+\ldots+c_{n_c} q^{-n_c}
\end{aligned}
\tag{1}
$$

AR refers to the auto-regressive part $A(q)y(\mathrm{t})$ where $y(\mathrm{t})$ the predicted variable is at time t and $y(\mathrm{t}-\mathrm{i}), i=1,\ldots,n_a$ are the actual values of variable y at past times $t-i,\ i=1,\ldots,n_a$. $C(q)e(\mathrm{t})$ is called moving average which acts as a moving average on white noise $e(\mathrm{t})$ and leads to colored noise. Therefore ARMA equation states that a realization $y(\mathrm{t})$ at time t depends on a linear combination of past observations $y(\mathrm{t}-\mathrm{i}), i=1,\ldots,n_a$ plus a moving average of $e(\mathrm{t})$, which is a white noise process characterized by zero mean and finite variance. In ARMA, least squares algorithm seeks for $\theta=[a_1,\ldots a_{n_a}, c_1,\ldots c_{n_c}]$ which are the ARMA model parameters. By calculating θ, we can obtain the prediction of $y(\mathrm{t})$ (wind speed) at time t, based on past (wind speeds) data $y(\mathrm{t}-1),\ldots,y(\mathrm{t}-n_a)$, model coefficients and MA part. In this model n_a and n_c (appeared in (1)) specify the orders of $A(q)$ and $C(q)$ polynomials respectively [17]. This model is shown as an ARMA(n_a, n_c) process. In this paper $y(\mathrm{t})$ is representative for the predicted hourly wind speed at hour t.

For developing an appropriate ARMA model, we should first determine the order of the AR and MA processes, n_a and n_c, respectively. This is the model identification phase which involves autocorrelation and partial autocorrelation analysis. It worth to note that the appropriate values of n_a and n_c can be selected by analysing the partial autocorrelation function for an estimate of n_a, and likewise using the autocorrelation function for an estimate of n_c. Then the unknown parameters of the model should be estimated using least squares algorithm. Finally the model should be checked with several diagnostics where the satisfactory result guarantees that the identified model performs properly.

3. Artificial Neural Networks

ANN is structured based on human brain. They are basically used to model the system inputs and outputs relationship. Neural networks have been applied for pattern recognition in historical observations, approximation and prediction. NNs have several advantages in comparison with the conventional forecasting methods which include error tolerance, learning, generalization and easy adaptability to online measurements [16].

In the learning process, NNs are trained using the training data where the network weights are learned by minimizing or maximizing a predefined criterion function. The NN is tested with the testing data that were not used during the training phase.

NNs have layers of interconnected processing elements called neurons. Every neuron performs computations on the received data and passes the results to another layer and finally the neurons in the output layer determine the output of the network. The training process adjusts the neuron weights to learn the desired input-output relation of the network.

There exist different types of NNs with simple or complex structures. NNs which are employed in this paper are explained in the following.

A) Multilayer Perceptron neural network

One of the most popular NNs is the multilayer perceptron (MLP) network. MLP neural network can approximate any nonlinear mapping with arbitrary precision. It can learn and adapt unknown information with a certain degree of fault-tolerance [18]. A simple MLP contains 3 layers of neurons named input, hidden and output layers. In the hidden layer, each node employs a nonlinear activation function. In order to train the MLP NN a supervised training method called back propagation (BP) is used.

In MLP, the neurons in the input layer receive the input vector $x = [x_1, ..., x_n]^T$ and apply the weighted input to the neurons in the hidden layer. The output of the hidden layer is the input for the output layer which provides the overall response of the network.

In a simple MLP NN with n input neurons, m hidden neurons, and one output neuron (shown in Fig. 1), the input layer receives the input and applies the weighted input to the hidden layer. All hidden layer nodes calculate their outputs as follows

$$z_j = f_H(\sum_{i=0}^{n} w_{i,j} x_i),\ i = 0,1,...,n;\ \ j = 1,...,m \tag{2}$$

where $\sum_{i=0}^{n} w_{i,j} x_i$ is the activation value of node j, $w_{i,j}$ is the connection weight from the input node i to hidden node j, $x_i, i = 1,...,n$ is the input with x_0 being the bias b_{HI} (with weight $w_{0,j} = 1$), z_j is the output of node j in the hidden layer. f_H called the activation function of every hidden node is usually a sigmoid function defined as follows

$$f_H(\sum_{i=0}^{n} w_{i,j} x_i) = 1 \Big/ 1 + \exp(-\sum_{i=0}^{n} w_{i,j} x_i) \tag{3}$$

Finally the output neuron receives the weighted $z_j, j = 1,...,m$ with a corresponding weights $w_{j,k}$ and calculates the NN output, y, as follows

$$y = f_o(\sum_{i=0}^{n} w_{j,k} z_j) \qquad (j = 0,1,2,...,m) \tag{4}$$

Where f_o is usually a linear activation function, $w_{j,k}$ is the connection weight from hidden node j to output node k (here $k = 1$), z_0 is the bias b_{HO} with weight $w_{0,k} = 1$.

All the connection weights and bias values are assigned with small random values initially, and then modified according to the BP training process. This supervised training method adjusts the connection weights by processing every data based on the output error comparing with the desired target. BP is a generalization of the least mean square algorithm. This method minimizes the network performance (usually described by mean square error (MSE)) by adjusting the weights. The algorithm repeats two phases, propagation and weight update. First an input vector is injected to the MLP, it is propagated forward through the network, layer by layer, until it reaches the output layer. The network output is then compared to the desired target output, using a loss function, and an error value is calculated for each neuron in the output layer. The error values are then propagated backwards, starting from the output, until each neuron calculates its associated error value which roughly represents its contribution to the original output. BP training uses these error values to calculate the gradient of the loss function. In the second phase, this gradient is used to update the weights, in order to minimize the loss function. After sufficient training epochs the MSE decreases gradually and becomes stable. Therefore after proper training, satisfactory results will be obtained in the testing phase.

B) Radial basis function neural network

The radial basis function (RBF) neural network is a three-layer feed-forward network illustrated in Fig. 2.The hidden layer neurons are multi-dimensional units where the dimension of every neuron is equal with the NN input unit dimension. The RBF basis functions only depend on the radial distance $\|x-c_j\|$ where different types of radial basis functions can be employed in the hidden neurons. Gaussian function defined below is usually used in RBF NNs.

$$z_j(\mathrm{x}) = \exp(-\frac{\|x-c_j\|}{\sigma_j^2}) \quad (5)$$

where c_j and σ_j are the center and the standard deviation of the j^{th} hidden neuron respectively, $\|.\|$ denotes the Euclidean distance, $z_j(\mathrm{x})$ is the output of the j^{th} neuron.

The output of the RBF NN is a linear weighted sum of all the hidden layer neurons outputs as follows

$$y = \sum_{j=1}^{m} w_{k,j} z_j(\mathrm{x}) \quad (6)$$

where $w_{k,j}$ is the weight between hidden neuron j and k neuron in the output layer and m is the number of hidden layer neurons. The RBF training contains two stages. First, the basis functions parameters are determined using random selection or a clustering method such a K-mean. Then the connection weights between the hidden and the output layers are adjusted using a supervised learning method for minimizing the least squares objective function such as gradient descent.

C) *Adaptive linear element neural network*

Adaptive linear element (ADALINE) is a simple single layer ANN. The structure of ADALINE NN is shown in Fig. 3. ADALINE network resemble perceptron where the only difference is that the neurons basis functions are linear. Therefore the ADALINE network is employed for the linear approximations of the functions. The network output is

$$y = Wx + b = \sum_{i=1}^{n} w_i x_i + b = \sum_{i=0}^{n} w_i x_i \quad (7)$$

where x_0 represents threshold bias b with weight $w_0 = 1$, $W = [w_1, \dots, w_n]$ represents the weight matrix corresponding to the one-column input vector $x = [x_1, \dots, x_n]^T$. As mentioned, ADALINE networks can only solve linearly separable problems. In order to train these networks, the least mean squared (LMS) or Widrow-Hoff learning rule can minimize the mean squared error (MSE) and search for the global minimum point in space. During the learning process, the LMS rules diminish MSE and find the optimal network weights.

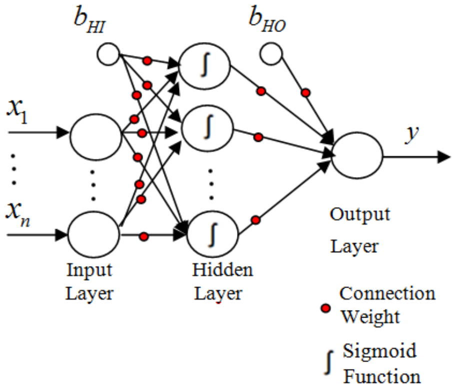

Fig. 1. Multilayer perceptron topology

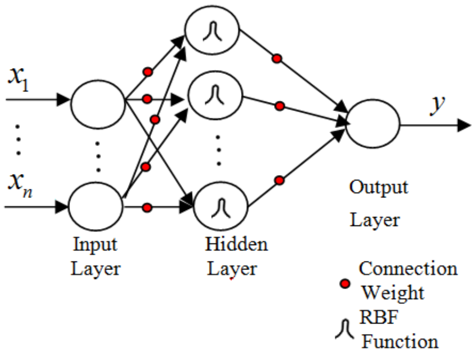

Fig. 2. RBF network topology

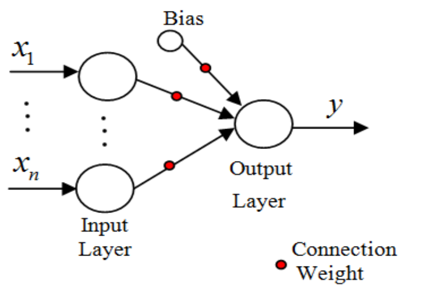

Fig. 3. Simple ADALINE network topology

4. Wind Speed Data

The existing one year wind speed database contains wind speeds, measured using a cup anemometer at 40 meters height of Binalood wind farm meteorological mast. The wind speed was continuously recorded and and stored as hourly values from January 1, 2010 to December 31, 2010. Fig. 4 illustrates the time series plot of the hourly wind speed at Binalood site, located in Khorasan-Razavi province of Iran, from January 1, 2010 to December 31, 2010.

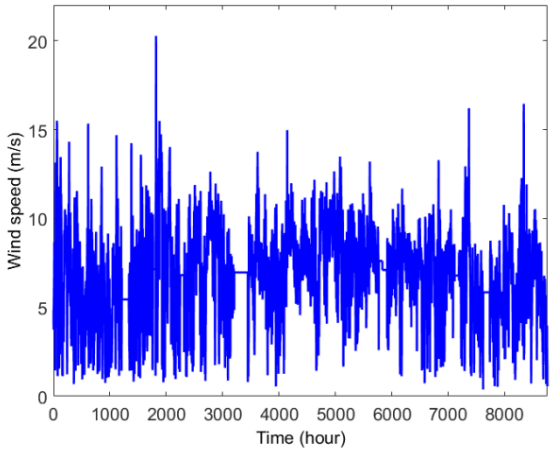

Fig. 4. Binalood Hourly wind speed at 40 meters height in year 2010

5. Model inputs and parameters

The wind forecast problem aims to find an estimate y_l of the desired wind y_l^d based on the n previous wind speed measurements $y_{l-1},...,y_{l-n}$.

In order to develop NN models which can forecast the hourly wind speed of Binalood, the number of input neurons shall be determined. Then the available data should be preprocessed. Based on trail and error in the three NN models, the number of neurons of the first (input) layer is chosen as 6 neurons ($n=6$). Therefore, the 6 previous observations of every target wind speed y_l^d (desired output), were preprocessed and formed in to one input vector. In MLP and RBF neural networks, the number of hidden layer neurons is selected as 5. The output layer of all three types of ANN models only contains one neuron representing the forecast value of next hourly averaged wind speed. The preprocessed dataset was further divided in to two subsets, training and testing datasets. With the 30% of datasets being separated as the testing dataset and other 70% were selected as the training dataset. ARMA model implies necessarily to know the relation that exists between the series and their lags. Autocorrelation function (ACF) and Partial autocorrelation function (PACF) are statistics tools that orient on the matter. Autocorrelation is the correlation of a signal with the delayed copy of itself as a function of delay. Partial autocorrelation gives the partial correlation of a time series with its own lagged values, after removing any linear dependence on all shorter lags. Generally, the ACF index approaches to zero at the cut-off point which is n_c as well as PACF where the last significant point is the order of n_a. Based on the analysis of ACF and PACF of the available Binalood wind data the orders of ARMA model is chosen as $n_a=3$ and $n_c=20$. Therefore ARMA (3, 20) is used for prediction in this paper. In order to evaluate an ARMA model which employs the same number of lags in comparison with the employed NNs, ARMA (6, 1) is also evaluated. In ARMA (6, 1), $n_a=6$ is chosen the same as the number of NNs input neurons and $n_c=1$ is determined by trial and error.

6. Simulation Results

For RBF NN, the hidden layer neurons centers are chosen randomly and the standard deviations are determined equal to 1. Fig. 5 depicts the predicted wind speed and real wind speeds of the testing set. The mean absolute error (MAE) and mean square error (MSE) of this prediction are 0.6423 and 0.8773 respectively. For MLP different learning rate constants are examined in the study where 0.125 provides the best performance. The MLP wind speed prediction result is shown in Fig. 6. The corresponding MAE and MSE are 0.6055 and 0.8820 respectively.

In ADALINE, maximum learning rate (MLR), 5.3045e-007, creates better results than other learning rates. Fig. 7 shows the estimated wind speeds in comparison with the actual wind speeds using ADALINE. Also the resulted MAE and MSE by the intended ADALINE are 0.7158 and 1.0998 respectively. The estimations by ARMA (6,1) and ARMA (3,20) in comparison with real wind speed testing set are depicted in Figs. 8 and 9 respectively. ARMA (6,1) resulted in 0.6116 and 0.8408 as MAE and MSE performance respectively and ARMA (3,20) leads to 0.6055 and 0.8299 as MAE and MSE respectively. Since the testing and training data sets in the 4 employed methods are the same, we can fairly compare their results. The obtained MAE and MSE for each method are gathered in Table 1. In one hand, if we compare the methods performance based on MAE criterion we conclude that MLP neural network and ARMA(3,20) provide the best results by 0.6055 MAE, on the other hand if methods performances are compared based on MSE, ARMA(3,20) provides the best result by resulting in MSE equal to 0.8299.

To overcome this duality we compare the concepts of MAE and MSE performance criteria. MAE and MSE definitions are as fallows

$$MAE=\frac{1}{T}\sum_{l=1}^{T}\left|y_l-y_l^d\right| ,$$
$$MSE=\frac{1}{T}\sum_{l=1}^{T}(y_l-y_l^d)^2 \quad (8)$$

where T is the total number of testing data, y_l and y_l^d are estimated speed and real speed at time l respectively. MAE measures the average magnitude of the errors in a set of forecasts. It is a linear score which means that all the individual differences are weighted equally in the average. MSE is a quadratic scoring rule which measures the average magnitude of the error.

Since the errors are squared before they are averaged, the MSE gives relatively high weight to large errors. This means the MSE is more useful when large errors are particularly undesirable. In the case of wind speed prediction where large errors result in considerable financial losses, MSE is a more reliable performance criterion. Therefore based on the resulted MSEs we can conclude that ARMA(3,20) by 0.8299 MSE provides the best result for Binalood hourly wind speed forecasting. MLP, after ARMA(3,20) and ARMA(6,1), provides the least MSE equal to 0.8820 and RBF results in 0.8773 MSE and ADALINE by 1.0998 MSE provides the poorest result in Binalood hourly wind forecasting.

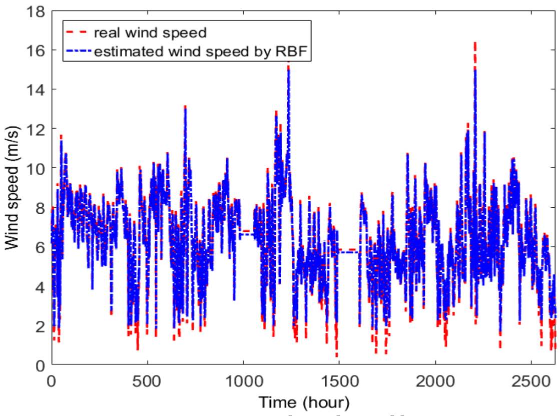

Fig. 5. Estimated wind speed by RBF

Table.1.
Comparison of employed methods performances

Method / Error	RBF	MLP	ADALNE	ARMA(6,1)	ARMA(3,20)
MAE	0.642	0.605	0.715	0.611	0.605
MSE	0.877	0.882	1.099	0.840	0.829

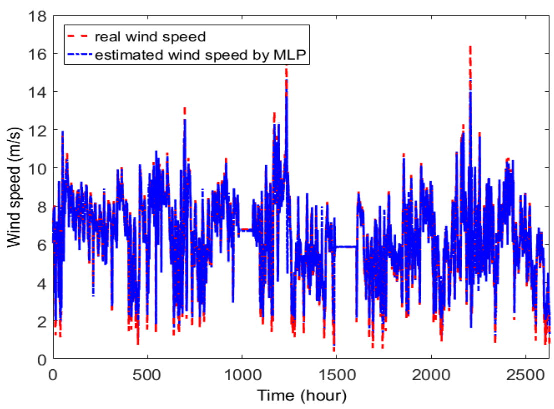

Fig. 6. Estimated wind speed by MLP

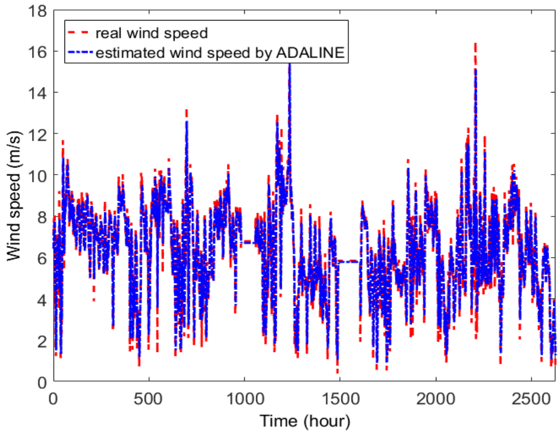

Fig. 7. Estimated wind speed by ADALINE

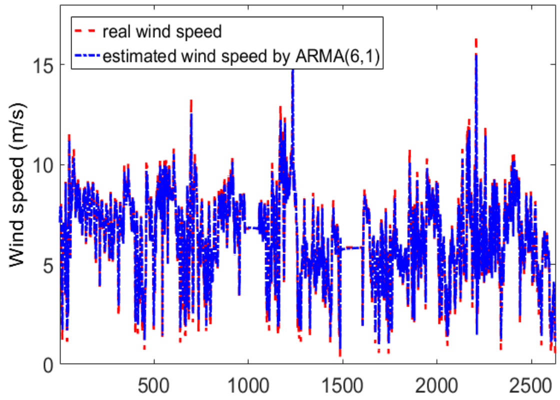

Fig. 8. Estimated wind speed by ARMA(6,1)

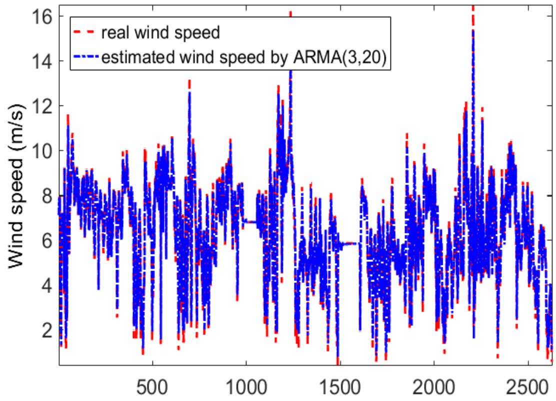

Fig. 9. Estimated wind speed by ARMA(3,20)

7. Conclusion

The main aim of this paper is to forecast hourly wind speed of Binalood site and investigate some available approaches on this problem to choose the most appropriate method. Since this forecasting is usually employed for predicting wind farm output power to maintain the power system balance and economic, choosing the best method is of great importance. Simulation results of four prediction methods, namely, MLP (multilayer perceptron), RBF (radial basis function), ADALINE (adaptive linear element) neural networks and ARMA time series model, indicate that the best accuracy is obtained by ARMA based on MSE criterion. Therefore we recommend ARMA as a superior method for Binalood hourly wind speed prediction among the four investigated methods. It should be mentioned that the results may be different in other sites or other short time prediction periods.

References

[1] H. Lund, "Large-scale integration of wind power into different energy systems", Energy, Vol.30 ,No.13 ,2005.

[2] S .Mathew, "Wind energy-fundamentals, resource analysis and economics", Berlin : Springer-Verlag, 2006.

[3] Y.K. Wu, and J.S. Hong, “A literature review of wind forecasting technology in the world,” IEEE Power Tech 2007, Lausanne, Switzerland, 2007.

[4] G. Xydis, C Koroneos,and M. Loizidou, “Energy analysis in a wind speed prognostic model as a wind farm sitting selection tool: a case study in Southern Greece.” Applied Energy, Vol.86, No.11, 2009.

[5] P. Pinson, H.A. Nielsen, H. Madsen, and G.Kariniotakis, “Skill forecasting from ensemble predictions of wind power,” Applied Energy, Vol.86, 2009.

[6] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z.Yan, “Are view on the forecasting of wind speed and generated power,” Renewable and Sustainable Energy Reviews, Vol.13, No.4, 2009.

[7] P.J. Luickx, E.D. Delarue, and W.D. D’haeseleer, “Considerations on the backup of wind power: operational backup,” Applied Energy, Vol.85, No.9, 2008.

[8] S.S. Soman, H. Zareipour, and O. Malik, “A Review of Wind Power and Wind Speed Forecasting Methods with Different Time Horizons,” North American Promoters Association & Conference, NAPA, 2010.

[9] J.K. Kaldellis, K.A. Kavadias, and A.E. Filios, “A new computational algorithm for the calculation of maximum wind energy penetration in autonomous electrical generation systems,” Applied Energy, Vol.86, 2009.

[10] D.A. Fadare, “The application of artificial neural networks to mapping of wind speed profile for energy application in Nigeria,” Applied Energy, Vol.87, No.3, 2010.

[11] J.M. Morales, R. Mínguez, and A.J. Conejo, “A methodology to generate statistically dependent wind speed scenarios,” Applied Energy, Vol.87, No.3, 2010.

[12] R. Kavasseri, and K. Seetharaman, “Day-ahead wind speed forecasting using f-ARIMA models” Renewable Energy, Vol.34, No.5, 2009.

[13] J. Torres, A. Garcia, M.D. Blas, and A.D. Francisco, “Forecast of hourly average wind speed with ARMA models in Navarre (Spain),” Solar Energy, Vol.79, No.1, 2005.

[14] K. Huang, L. Dai, and S. Huang, “Wind Prediction Based on Improved BP Artificial Neural Network in Wind Farm,” International Conference on Electrical and Control Engineering, Wuhan, China, 2010.

[15] H.G.Beyer, T. Degner, J. Hausmann, M. Hoffmann, and P. Rujan, “Short-term predictions of wind speed and power output of a wind turbine with neural networks,” 2nd European Congress on Inteligent Techniques and Soft Computing, Aachen, Germany, 1994.

[16] A. More, and M.C. Deo, "Forecasting wind with neural networks. Marine Structures", Vol.16, No.1, 2003.

[17] L. Ljung, "System identification: Theory for the user", 2nd ed., New Jersy: Prentice Hall PTR, 1999.

[18] B. Widrow, and M.A. Lehr, “30 years of adaptive neural networks: perceptron, madaline, and back propagation,” Proceedings of the IEEE, Vol.78, No.9, 1990.

[19] P.J. Brockwell, and R.A. Davis, "Introduction to time series and forecasting", NewYork, Springer, 2002.

[20] S.Li, H. Wang, and Y. Tian, “Wind speed forecasting based on fuzzy-neural network combination method,” 27th Chinese Control and Decision Conference (CCDC), 2015 .

[21] I. Colak, S. Sagiroglu, M.Yesilbudak, E. Kabalci, and H. I. Bulbul, “Multi-Time Series and -Time Scale Modeling for Wind Speed and Wind Power Forecasting, 4th international conference on renewable energy research and applications,” Palermo, Italy, 2015.

[22] M. A. Duran, and U. B. Filik, “Short-Term Wind Speed Prediction Using Several Artificial Neural Network Approaches in Eskisehir,” International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Madrid, Spain, 2015.

[23] L. Chen, and X. Lai, “Comparison Between ARIMA and ANN models Used in Short-term Wind Speed Forecasting,” Asia-Pacific Power and Energy Engineering Conference (APPEEC), Wuhan, China, 2011.

[24] R.S.Tarade, and P.K.Katti, “A comparative analysis for wind speed Prediction,” International Conference on Energy, Automation, and Signal (ICEAS), Bhubaneswar, Odisha, India, 2011.