International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 3, No. 1, 2016, pp. 103-121.
ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

Dynamic Modeling of Wind Speed and Temperature Using Nonlinear Auto Regressive with eXogenous (NARX)

Elham Alipourtarzanagh[a], Mehrdad Boroushaki [b]

a Department of Energy Systems Engineering, Islamic Azad University-Science and Research Branch, Tehran, Iran.

b Department of Energy Engineering, Sharif University of Technology, Azadi Street, Tehran, P.O. Box 14565-114, Iran.

Abstract

The current research aims to develop a dynamic neural network model in order to forecast daily wind speed and air temperature. A Nonlinear Auto Regressive with eXogenous (NARX) structure is being used for wind speed and air temperature forecasting in two cities in Iran. The effective inputs and outputs of the model have been found through trial and error. Temperature and wind speed are found as functions of dew point, station pressure, cloud cover, humidity and visibility. Due to a variety of delays in the NARX network and hidden layer neuron numbers, the best structures are developed through conducted trials. Adaptive learning rate and batch learning technique have been used in the training phase. The results show that the developed model can forecast one day ahead wind speed and temperature with a Mean Absolute Error=0.91 (m/s) and Mean Absolute Error = 1.80 (°C), respectively. The forecasted wind speed and temperature conform well to the observed data.

Key words: wind speed, temperature, forecasting, NARX, Recurrent Neural Network

Nomenclature

CC	Cloud Cover
d	Desired value
DP	Dew point temperature
G	Number of Transient
H	Humidity
L	Layer number
m	Weight on the "memory" of previous step
n	Iteration number
O_i	Activation function of unit i
r	Number of day
SP	Station Pressure
Temp	Temperature
T_r	Number of entire data
TTSE	Total Transient Squared Error
V	Visibility
w_{ij}	Weight from unit j to unit i
WS	Wind Speed
y_i	Activation function of unit i
y_k	Actual value of unit k
Z^{-1}	Delay unit
η	Learning rate
δ	Error signal

Subscripts

i	Input unit
j	Hidden unit
k	Output unit

Introduction:

Great attention has been devoted to the topic of wind speed and temperature forecasting which are integral in many fields such as energy management, meteorology and agriculture. Wind is one of the clean energy resources that may lead to reduced fossil fuel dependency and the development of sustainable energy (Jong et al., 1996). The discontinuous and stochastic nature of wind energy affects the power grids. Therefore, it cannot be propounded as a controllable energy resource which can be scheduled and planned. Additionally, the cumulative energy input of several wind power plants into the electric grid may cause fluctuations in the power systems. Therefore, wind speed forecasting is critical for the security of wind power integration, system planning, and dynamic control of wind turbines. As a result, reliable wind power forecasting systems are required for the optimization of the produced electricity with respect to market requirements i.e. unit commitment, economic dispatch, participation in the electricity markets, etc. (Giebel et al., 2011).

In the last decade, great efforts have been made to develop wind speed and power forecasting models. Accurate wind power forecasting methods are vitally important in balancing supply and demand power. Such forecasting is required to maximize the integration of wind power penetration into an electricity system (Venayagamoorthy et al. 2012). Improved accuracy of wind power forecasts leads to higher grid reliability and attractive cost savings (Dobschinski et al., 2008). The conducted studies in wind speed forecasting can be divided into short term forecasting and long term forecasting. Long term forecasting considers the time scales of wind speed data and the aims of the studies while short term forecasting leads to minimizing the scheduling errors which impacts on grid reliability and additional market based costs. In addition long term forecast is important for site selection, wind mill planning, and the selection of an optimal size of the wind machine for a specific site (Hu et al., 2013).

The methods which are used in forecasting wind speed can be considered in order to categorize the models. Hui Liu et al. have classified wind speed prediction methods into three main categories: 1) Physical Methods, 2) Statistical Methods, and 3) Intelligent Methods (Liu et al., 2012). Physical models use physical concerns such as terrain, obstacle, pressure, and temperature to estimate the future wind speed (Lei et al., 2009). Atmospheric physics containing wind power meteorology can be modeled by Numerical Weather Prediction (NWP) models which are general purpose models. These models are not specifically for wind energy applications and can reach to the order of two kilometers resolution in their non- hydrostatics version while for micro siting and wind energy applications higher resolutions are commonly mandatory. Furthermore, NWP models often display systematic errors in the forecasting of certain meteorological parameters, such as wind speed and direction, notably near the surface (Cassola and Burlando, 2012). In statistical models, the time series models are widely used due to their computational simplicity (Liu at al., 2012), (El-Fouly et al., 2008). Time series based models use only historical wind speed data recorded at the site to build statistical models from which forecasts are derived. These forecasting methods have confidence in the classical Box– Jenkins methodology, which employs a general class of models such as the Auto-Regressive Moving Average (ARMA), or Auto- Regressive Integrated Moving Average (ARIMA) models to obtain forecasts (Kavasseri and Seetharaman, 2009). Intelligent algorithm modeling, such as Artificial Neural Networks (ANN) and Support Vector Machines

(SVM), makes a non-linear function of high dimension to fit the historical wind speed and its influencing factors usually with a minimization of the training error (Liu et al. 2014).

ANNs are used for time series forecasting. There are various ANN model structures available for forecasting, but the feed forward neural network has been the most commonly used neural network model (Parsaei and Salehi, 2015). SVMs are a widely used approach for statistical analysis applications such as classification, data mining, and forecasting (Parsaei at al. 2016). The bedrock of SVM, as opposed to the ANN approach, is that this approach codifies the statistical learning problem as a quadratic programming model with linear constraints (Shi et al. 2012). Hybrid models are always expected to have a higher accuracy than forecasting models which only use single algorithms(Liu at al. 2013), these hybrid methods have been utilized for wind speed forecasting in several researches (Gnana and Deepa, 2013), (Guo et al. 2011). The time series models and intelligent models were combined to forecast wind speed better than other data processing techniques such as the Genetic Algorithm (GA) (Liu et al., 2014). In addition some efforts have been made for temperature forecasting (Smith et al., 2007). (Venkadesh et al. 2012) improved the air temperature forecasting accuracy of the ANN models by employing a GA to refine input data selection (Parsaei et al. 2016).

In this research, the NARX structure is developed for modeling wind speed and air temperature. Static ANN modelling (Yeh et al. 2014) and recurrent neural network (Barbounis et al., 2006) have already been used in wind speed forecasting. However, since implementing NARX is much easier than conventional recurrent neural networks, the current work has accomplished this forecasting via NARX for the first time. Furthermore, the nonlinearity order of the model input and output parameters have been found truly by a trial and error approach. For modeling a steepest descent algorithm the batch learning technique is used. Among all available data for each city we trained and tested the ANN for duration of the last 20 years, 70% of the data (14 years) were employed for training, and remaining 30% of the data (6years) were adopted for testing and model verification. The resulting model can be used for wind speed and air temperature forecasting in different applications.

Artificial Neural Networks

An Artificial Neural Network (ANN) is able to learn and memorize a data structure then simulate this structure accordingly. ANN is able to learn key information patterns within a multidimensional information domain (Kalogirou, 2001). The main work of ANNs is extracting information from a data structure by interpolation or extrapolation between the data. They consist of a great number of processing elements (neuron), connected to each other. The network usually consists of an input layer, some hidden layers, and an output layer (Kalogirou, 2000). Mehrian et. al. did a comprehensive investigation on the calculation of total strain energy as well as the thermoelastic of material before any fracture or corrosion grow in the advanced materials. (Mehrian et. al., 2013, 2014, 2015, 2016). These networks are called Multi-Layer Perceptron (MLP) (Fig. 1).

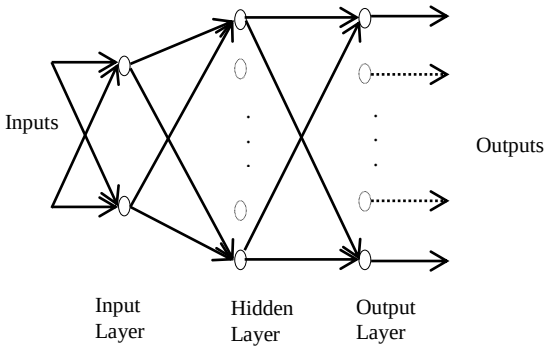

Fig. 1. A MLP neural network

Each input is multiplied by a connection weight. Then a transfer function acts on it (multiplied input by its connected weight) and generates an output. Network biases represent relationships between inputs and outputs easier than networks without biases (Palau et al. 1999). In ANN, the knowledge lies in the interconnection weights between neurons therefore, an important subject of a neural network is the training step.

In most applications of MLP, the weights are determined by means of the error back-propagation algorithms (EBP), which minimizes a quadratic cost function with a gradient descent method. During the training phase, the weights are successively adjusted based on a set of inputs and the corresponding set of desired output targets.

First, the inputs are presented to the network and propagated forward to determine the resulting signal at the output neurons. Desired outputs of the neural networks are the measured meteorological values that ANN tried to decrease differences between them and computed outputs. The differences between the computed output vectors and the desired output targets represent an error that is back-propagated through the network in order to adjust the weights. This process is repeated and the learning continues until the desired degree of accuracy is achieved (Haykin, 1999). According to the back-propagation algorithms, when an input is presented to the network, the activation of each neuron is determined by

$$O_i = y_i(\sum_{j=0}^{J} w_{ij}\, O_j) \qquad (1)$$

Where O_i is the output of unit i, y_i is the activation function of unit i, w_{ij} is the weight from unit j to unit i, and J is the total number of inputs (excluding the bias) applied to neuron i. The synaptic weight w_{io}(corresponding to the fixed input) equals the bias applied to the neuron i. Back-propagation is then invoked to update all the weights in the network according to the following rule.

$$\Delta w_{ij}(n+1) = \eta(\delta_i . O_j) + m.\Delta w_{ij}(n) \qquad (2)$$

where n is the iteration number,η is the learning rate,δ_i is the error signal for unit i, and m is the momentum factor. The error signal δ_k for an output unit k is calculated through the difference between desired value d_k and actual value y_k for that unit, while the error signal δ_h for a hidden unit h is a function of the error signals of those units in the next layer connected to unit h and the weights of those connections. The two parameters η and m represent respectively an adjustment of step size and a weight on the "memory" of previous steps. Assuming that η and m are appropriately chosen the back-propagation process, will generally converge to a minimum that satisfies the criterion imposed by the user, which usually renders the sum of the squares of the error of the output signals, $\sum_k (d_k - O_k)^2$ less than a predetermined value. In this research, the momentum factor was set to zero and the bias of all neurons was set to +1, an adaptive learning method for determining the amount of η in each iteration was used. According to this method, once the amount of error rises, the η parameter will decrease and vice versa. The training was carried out by the batch learning method, in which the entire plant data sets $(1,\ldots,T_r)$ during a transient was used for learning, until the total transient output error, $\sum_{r=1}^{T_r} E(n,r)$ reached below a certain value. T_r is the number of the entire data sets in a transient. Thus, the signal vector applied to the input layer of the MLP, consists of a data window made up of present and past values of the plant inputs representing exogenous inputs originated from outside the network and delayed values of the model outputs on which the model outputs are regressed. This recurrent network is named the NARX model (Nonlinear Auto Regressive with eXogenous inputs). The dynamic behaviour of the NARX model is described by

where F is a nonlinear function of its arguments (Boroushaki,2003). (Fig. 2).

$$y(r+1) = F(y(r), \ldots, y(r-q+1), u(r), \ldots, u(r-p+1)) \tag{3}$$

The model suggested in this study for wind speed and temperature forecasting is a Multi Input Multi Output (MIMO) model in which outputs of the model are considered as functions of dew point, station pressure, cloud cover, visibility, humidity and delays of inputs and outputs. Identification of such a system requires more powerful learning methods, compared to ordinary methods used for neural networks; otherwise, the dynamic behavior of the model cannot be adequately explored. In the batch learning method the neural network is trained by current and historic meteorological data. The simulation results show that in this method, a neural network has better ability to learn in the dynamic models.

Batch learning together with the following heuristic method for calculation of the network weights, which resulting results in considerable improvement in responses, is used in the current study. If the number of entire data sets in a transient is equal to T_r , then the total transient squared error (TTSE), with scaled $d_k(t)$ and $y_k(n,r)$ values in a batch learning, will be

$$TTSE = \sum_{r=1}^{T_r} E(n,r) = \frac{1}{2} \times \sum_{r=1}^{T_r} \sum_{k=1}^{K} (d_k(r) - y_k(n,r))^2 \tag{4}$$

The updates network weights, between the jth neuron in layer $l-1$ and the ith neuron in layer l is (after any iteration)

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \tag{5}$$

where n is the iteration number (an iteration contains all T_r data sets during a batch learning) and $\Delta w_{ij}(n)$ the weight variation in the nth iteration, which can be calculated as follows:

$$\Delta w_{ij}(n) = \frac{\sum_{r=1}^{T_r} \Delta w_{ij}(n,r)}{T_r} \quad (6)$$

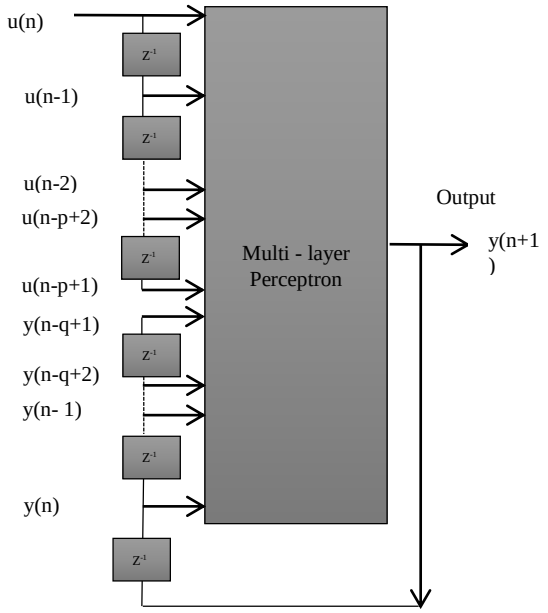

Fig. 2 . NARX structure

where $\Delta w_{ij}(n, r)$ is the weight difference, calculated for any data sets at the nth iteration during an instantaneous learning with steepest descent rule

$$\Delta w_{ij}(n, r) = -\eta \times \frac{\delta E(n,r)}{\delta w_{ij}(n,r)} \quad (7)$$

Iteration between (5)–(7) on NARX weights will reduce the TTSE in (4). The learning rate η in NARXs has been selected as a variable during training iterations and among network layers. It is recalculated at any iteration, in such a manner that the average value of weights variations in any layer will be equal during the training. In our simulation, the range of its variation was $5 \times 10^{-4} < \eta < 10$. This method results in a uniform distribution of the learning pattern in all network layers and increases the network learning capability and convergence.

Network training is very sensitive to initial network weights order. We selected initial network weights randomly in a magnitude of 10^{-1}. Selecting a higher initial weights order can cause saturation in the neurons output and consequently no network learning, while a lower initial weights order would extend the required training time.

Another problem to be solved is how to regulate the weights during the learning process, since learning different transients by network regulates the weights vector in different directions (resulting in a very large weights vector) which may cause the loss of previous learning. This shortcoming could be overcome by the addition of a weight difference Δw_{ij} to the network weight, defined as follows:

$$\Delta w_{ij} = \frac{\sum_{g=1}^{G} \Delta w_{ij}(g)}{G} \quad (8)$$

Where each transient includes all meteorological data during one year, G is the number of transients and $\Delta w_{ij}(g)$ is the weight variation during each transient, computed from (6).

Equations (4)–(8) must be used iteratively, until the sum of the TTSEs from all transients reaches below a certain value (Gandomkar and Kiarasi, 2006).

Meteorological Data

The required data for the current study has been collected from Iran Meteorological Organization (IRIMO). Wind speed and visibility data were provided every 3 hours a day and the rest of the data were on a daily average basis. Among the applicable cities for wind farm installation (Gandomkar and Kiarasi,2006),we have chosen Tabriz (38.5 N, 46.17E, Elevation 1364m) and Zahedan (29.28N, 60,53E, Elevation 1370m) for modelling. The provided data for these two cities were collected from 1993 to 2012, which has been used for the network training and test.

Before implementing the data into the neural network the available data of Dew Point(DP), Station Pressure(SP), Cloud Cover (CC), Temperature(Temp), Wind Speed(WS), Humidity (H) and Visibility(V) needs to be normalized in order to be scaled between -1 and +1. For instance, equation (9) is used for normalizing DP as follows:

$$DP_{normal} = \left(2 \times \left(\frac{DP - DP_{min}}{DP_{Max} - DP_{min}}\right)\right) - 1 \quad (9)$$

where DP_{normal} is the normalized value of DP, DP_{min} and DP_{max} are the minimum and maximum values of DP, respectively. In order to normalize the rest of the data the same equation has been used.

Optimal design of the NARX structure

Figure 3 shows the NARX structure designed for wind speed and temperature forecasting. This model is of a Parallel Identification Model since delays of outputs of the model are fed back to the NARX inputs (Hykin, 1999). Finding the most effecting input parameters of the proposed NARX model for wind speed

has been done based on a trial and error method to capture the wind speed dynamic system. In Figs. 4 to 6, different meteorological parameters have been selected as inputs and outputs of the NARX model, while Fig. 6 shows the lowest forecasting error. Based on this method, main inputs of NARX model are Dew Point, Station Pressure, Cloud Cover, Visibility and Humidity parameters.

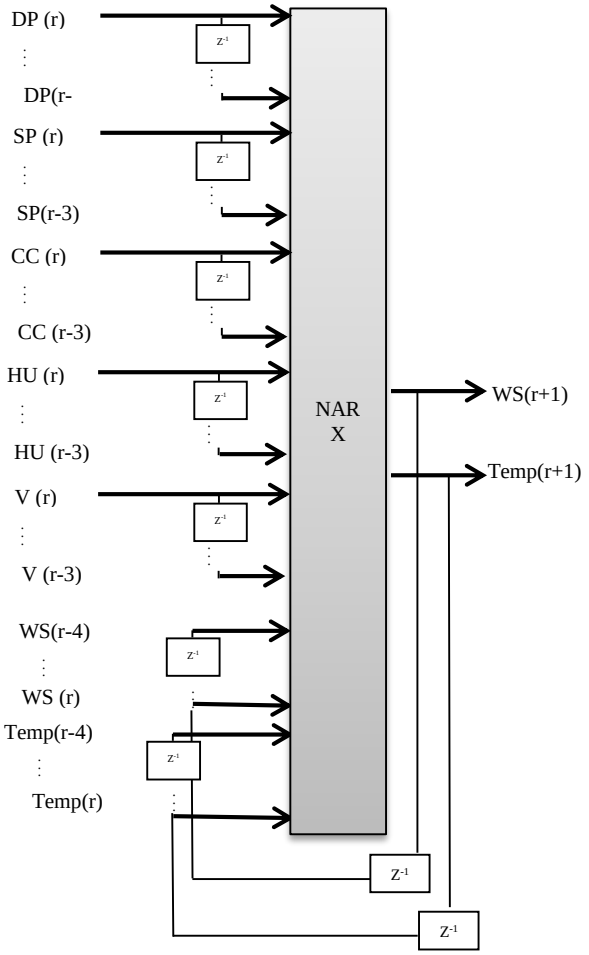

Fig. 3. NARX Structure designed to forecast the wind speed and

Nonlinearity order of the wind speed dynamic system in inputs and outputs, which is directly shown by the number of delay lines, has been found through trial and error, too. In Fig. 7 to 11 different numbers of delay lines were implemented, while Fig. 9 shows the lowest forecasting error. The number of input layer neurons depends on the delays in output and input parameters (p and q). In the current study, the number of input layer neurons with p=3 and q=5 is equal to

$$number\ of\ input\ layer\ neurons = (p+1) * (number\ of\ input\ parameters) + (q) * (number\ of\ output\ parameters) \quad (10)$$

where number of input parameters is equal to 5 and number of output parameters equals to 2 (see Fig. 6).

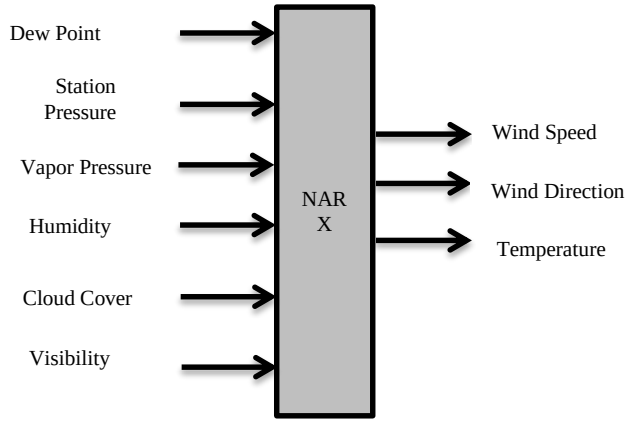

Fig. 4. Finding the most effecting input and output parameters of the proposed NARX model- step1

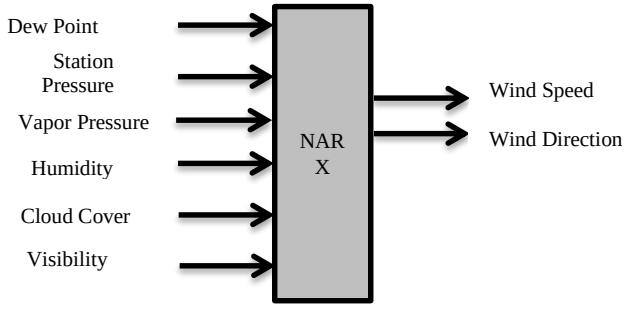

Fig. 5. Finding the most effecting input and output parameters of the proposed NARX model- step 2

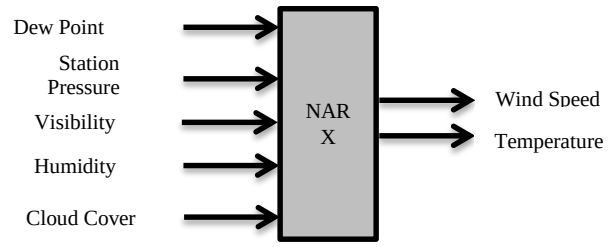

Fig. 6. Finding the most effecting input and output parameters of the proposed NARX model- step 3

The dynamic behaviour of wind speed and temperature can be described by a discrete nonlinear function as follows.

$$f\big(ws(r+1)\big) = f_p\big(ws(r), \dots, ws(r-q+1), Temp(r), Temp(r-q+1),\ DP(r), \dots, DP(r-p+1), SP(r), \dots, SP(r-p+1), CC(r), \dots, CC(r-p+1), H(r), \dots, H(r-p+1), V(r), \dots, V(r-p+1)\big)$$

$$f\big(Temp(r+1)\big) = f_p\big(ws(r), \dots, ws(r-q+1), Temp(r), Temp(r-q+1),\ DP(r), \dots, DP(r-p+1), SP(r), \dots, SP(r-p+1), CC(r), \dots, CC(r-p+1), H(r), \dots, H(r-p+1), V(r), \dots, V(r-p+1)\big) \quad (11)$$

where r presents day intervals.

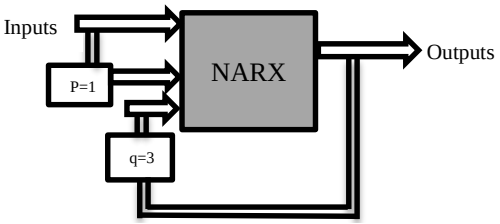

Fig. 7. Finding nonlinearity order of the wind speed system- step 1

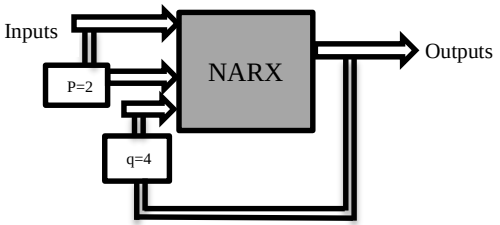

Fig. 8. Finding nonlinearity order of the wind speed system- step 2

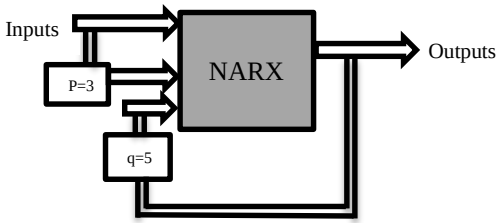

Fig. 9. Finding nonlinearity order of the wind speed system- step 3

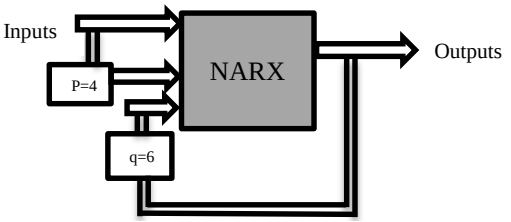

Fig. 10. Finding nonlinearity order of the wind speed system- step 4

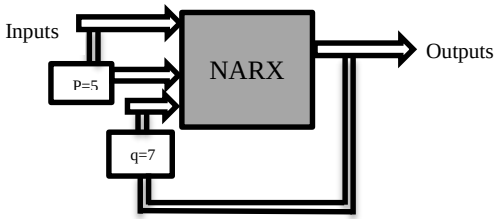

Fig. 11. Finding nonlinearity order of the wind speed system- step 5

During the training phase, error is calculated at the end of each learning iteration in order to reach under a certain desired value. TTSE for this study is:

$$TTSE = \sum_{r=1}^{T_r} E(n,r) = \frac{1}{2} \times \sum_{r=1}^{T_r} ((WS_d(r) - WS_k(r))^2 + (Temp_d(r) - Temp_k(r))^2) \qquad (12)$$

where WS_d, WS_k, $Temp_d$ and $Temp_k$ are measured wind speed, forecasted wind speed output of the model, measured temperature, and forecasted temperature output of the model at the n th day, respectively.

1. Training the NARX Network

A NARX structure with two hidden layers and 31 input neurons, 67 neurons in the first hidden layer, 32 neurons in the second hidden layer, and 2 output neurons was used for Tabriz, and for Zahedan these numbers were 31, 63, 40 and 2, respectively. The number of the first and second hidden layer neurons was found through trial and error.

For the purpose of simulation data measured from 1993 till 2006 was used for the network training and data from year 2007 to year 2012 was used for testing. The average number of iterations to reach the desired result is iteration=250. The training has been done for 14 years. Equation number (4) is used for calculating TTSE for each year where the input data includes data gathered from 365 days. The weights have been regulated using Equation (8). Fig. 12 shows the changes of TTSE during the training phase. As it can be seen in Fig. 12, changes of TTSE in each transient are not stable at the beginning but as the network training continues it gradually becomes stable and at the end it reaches the desired value.

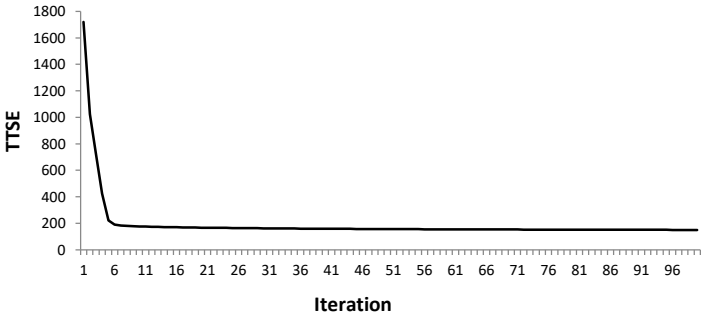

Fig. 12. Changes of TTSE vs. iterations

Learning rate η is set toη = 1.1, initially. Since the adaptive learning method has been applied, learning rate η is adapted to TTSE by the coefficient of 0.999 and 1.001. Fig. 13 shows the changes of η through the iterations. Connecting weights between the layers are adjusting during the training phase that shows the learning process is going well.

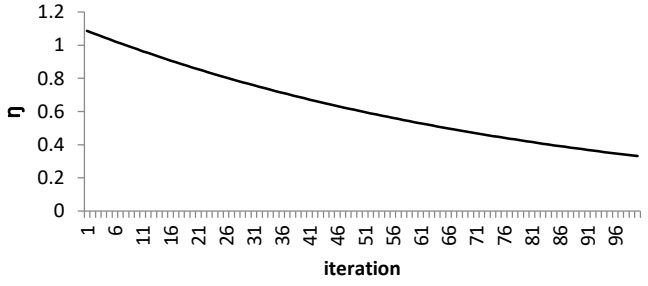

Fig. 13. Changes of η vs. iterations

2. *Simulation* Results

To compare the simulation results, the following equations have been used to calculate the error of wind speed prediction. The same equations are used for temperature forecasting.

$$Root\ Mean\ Square\ Error\ (RMSE) = \sqrt{(\sum_{r=1}^{r}(WS_r - \overline{WS}_r)^2)/r} \quad (13)$$

$$Mean\ Absolute\ Percentage\ Error(MAPE) = \frac{1}{r}\sum_{r=1}^{r}\left|\frac{WS_r - \overline{WS_r}}{WS_r}\right| \times 100 \quad (14)$$

$$Mean\ Absolute\ Error\ (MAE) = \frac{1}{r}\sum_{r=1}^{r}\left|WS_r - \overline{WS_r}\right| \quad (15)$$

$$Mean\ Square\ Percentage\ Error(MSPE) = \frac{1}{r}\sum_{r=1}^{r}(\frac{WS_r - \overline{WS_r}}{WS_r})^2 \times 100 \quad (16)$$

where WS is the measured data and $\overline{WS}$ is the forecasted value.

Figures 14 and 15, show the forecasted and measured values of wind speed and temperature for Tabriz in one year selected randomly from the 6 years of test data. In Fig. 16 and 17, the measured and forecasted wind speed and temperature can be seen for Zahedan during one year selected randomly from the 6 years test data.

The errors presented in Table I shows that the model forecasts wind speed and temperature with acceptable accuracy. The results reveal that the NARX model is forecasting wind speed with more accuracy for Tabriz while the accuracy of the model in temperature forecasting in Zahedan is better than Tabriz.

TABLE 1. MSPE, MAPE, RMSE and MAE for Tabriz and Zahedan over one year

		MSPE (%)	MAPE (%)	RMSE	MAE
Tabriz	***Wind Speed***	***11.3***	***24.2***	***1.198***	***0.862***
	Temperature	***156.3***	***43.5***	***2.384***	***1.869***
Zahedan	***Wind Speed***	***29.7***	***29.7***	***1.423***	***0.970***
	Temperature	***276.2***	***12.7***	***2.187***	***1.723***

Discussion

In order to reach the best possible results in proposed NARX model several modifications have been applied through the training phase. For instance, batch learning and the adaptive learning rate have been implemented. Table II. compares accuracy of the forecast model applied in the current study with the static MLP (Guo et al. 2011), (Smith et al., 2007) and linear time series (El-Fouly et al., 2008).

As can be seen in Table II, MAE and MAPE errors of the NARX structure is less than the static MLP with raw input, while additive and multiplicative data processing reached a smaller MAPE value than NARX. Comparisons of NARX results with the best results of the linear time series based model (El-Fouly et al., 2008) for 24 hours intervals shows much better performance of the NARX model.

In addition, results of the static MLP for air temperature forecasting in 12 hours intervals horizon and those of NARX for 24 hours reveals a meaningful improvement in NARX temperature forecasting (Smith et al., 2007).

On the other hand, simplicity of the proposed method is an advantage of this model. Comparing the proposed NARX model with the hybrid models (Liu et al., 2013), (Yeh et al., 2014) and models which refine the input (Guo et al. 2011) shows the easy applying of the proposed model.

TABLE 2. Comparison of the NARX forecasting results with Static MLP and Linear Time Series methods

Forecasted Parameter	*Method*		*MAPE (%)*	*MAE*	*RMSE*	*Horizon*
Wind Speed	*Static MLP (Guo et al. 2011)*	*Raw*	*59.4*	*N/A*	*N/A*	*24 hours*
		Additive	*23.03*	*N/A*	*N/A*	*24 hours*
		Multiplicative	*21.13*	*N/A*	*N/A*	*24 hours*
	Linear Time Series (El-Fouly et al.,2008)		*N/A*	*1.267 (m/s)*	*1.589*	*24 hours*
	NARX	*Tabriz*	*24.2*	*0.862 (m/s)*	*1.198*	*24 hours*
		Zahedan	*29.7*	*0.970 (m/s)*	*1.423*	*24 hours*
Temperature	*Static MLP (Smith et al., 2007)*		*N/A*	*2.83 °C*	*N/A*	*12 hours*
	NARX	*Tabriz*	*43.5*	*1.87 °C*	*2.38*	*24 hours*
		Zahedan	*12.7*	*1.73 °C*	*2.18*	*24 hours*

Total time needed for the training phase and test phase for Tabriz is 2.11 minutes and for Zahedan is 3.08 minutes on a computer with an Intel Core 2 duo processor P8700 (2.35 GHz) with MATLAB programming.

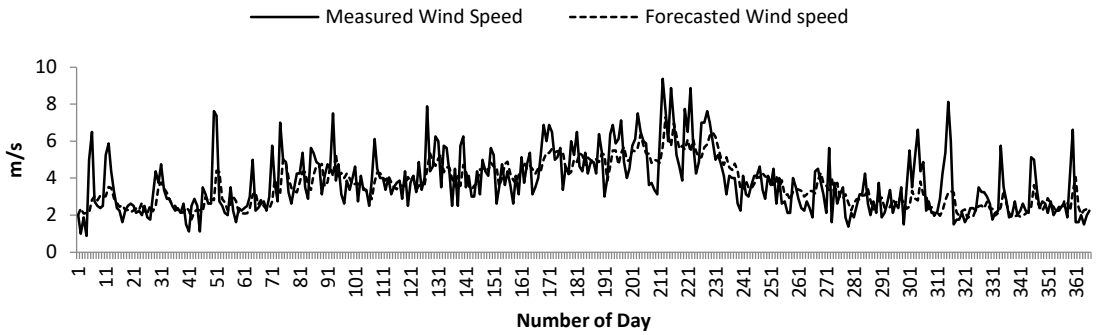

Fig. 14. Measured and forecasted average daily wind speed in Tabriz during year 2008

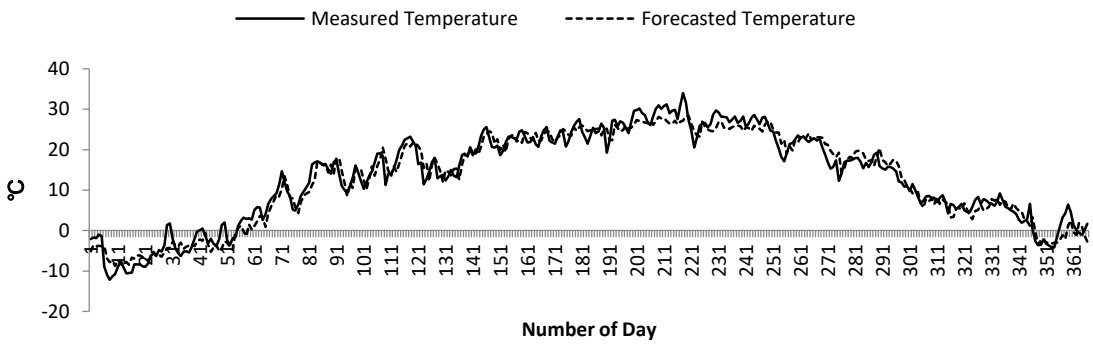

Fig. 15. Measured and forecasted average daily Temperature in Tabriz during year 2008

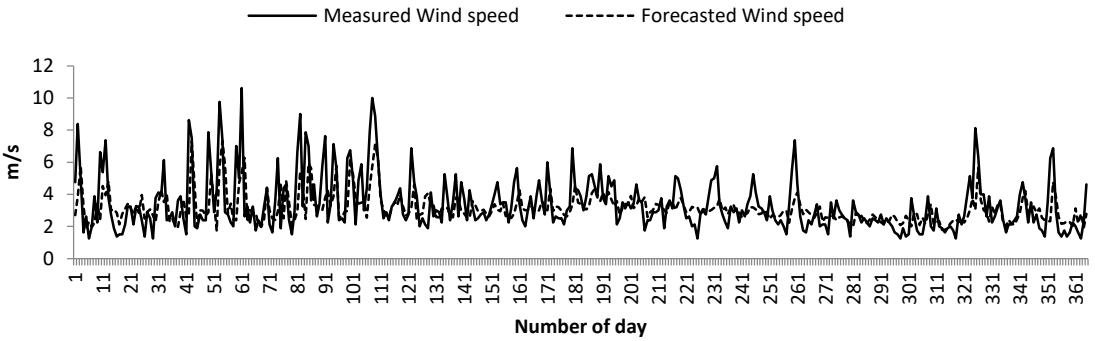

Fig. 16 Measured and forecasted average daily wind speed in Zahedan during year 2009

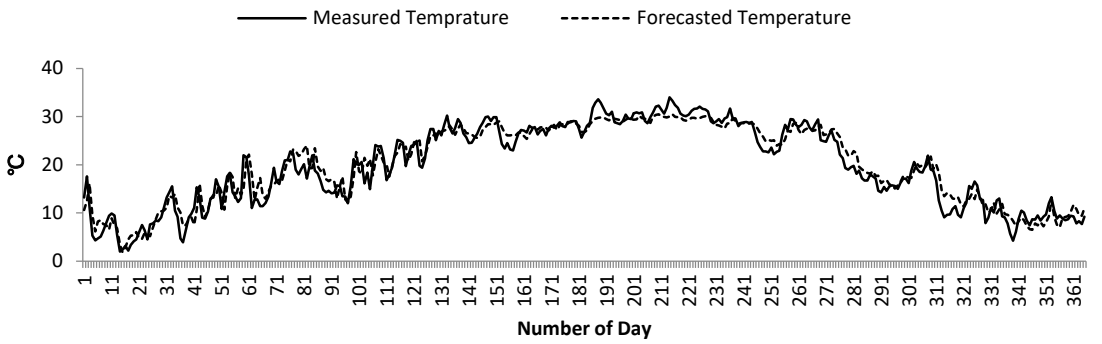

Fig. 17 Measured and forecasted average daily Temperature in Zahedan during year 2009

Conclusions

This research proposes a Nonlinear Auto Regressive with eXogenous inputs, NARX neural network for modelling wind speed and temperature forecasting. Static ANN modelling and recurrent neural network have already been used in wind speed forecasting. However, since implementing NARX is much easier than conventional recurrent neural networks, the current work has accomplished this forecasting via NARX for the first time. The NARX model, in the form of a parallel with two hidden layers, was used for two cities inputs of dew point, station pressure, cloud cover, humidity and visibility. The model has been designed to be relatively simple and easily simulated in contrast to other models such as hybrid models which have to deal with adjustment of input data.

In order to validate the model, real measured data of a duration of six years in the cities has been used. Learning methods and the general structure of NARX are not related to the model (the city); although, considering the topologic differences between the selected cities (Tabriz and Zahedan), the dependency of the number of hidden layer neurons is reasonable and necessary. While in hybrid models this dependency of the forecasting system (order of dependence) is much greater than the NARX model.

As it has been illustrated above, we have applied different structures of the NARX network for each city to find the optimal structure. Although the number of delay units in inputs and outputs of the NARX model are the same for both cities, the numbers of hidden layer neurons are different. This problem arising from different dynamics governing on wind and temperature in two cities. Furthermore, even for the best structure, errors vary for the two cities.

The results from this research can be used in energy management and wind farm power plant and power grids, as well as other fields such as meteorology and agriculture.

References:

Barbounis,G.T. & Theocharis, J.B.& Alexiadis, M.C.& Dokopoulos, P.S. (2006). Long-term wind speed and power forecasting using local recurrent neural network models", IEEE transactions on energy conversion, 21, pp. 273-284, 2006.

Boroushaki, M. & Ghofrani, M. B. & Lucas, C. & Yazdanpanah, M. J. (2003). Identification and control of a nuclear reactor core (VVER) using recurrent neural networks and fuzzy systems, IEEE Transactions on nuclear science, 50, pp. 159-174.

Cassola, F. & Burlando, M. (2012). Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output, Applied Energy, 66, pp. 99-154.

Dobschinski,J. & Wessel, A.& Lange, B.& Rohrig, K.& Bremen, L. v. & Saint-Drenan, Y.M. (2008). Estimation of wind power prediction intervals using stochastic methods and artificial intelligence ensembles, Deutsche Windenergie Konferenz DEWEK, Bremen, Germany

El-Fouly, T. H. M. & El-Saadany, E. F. & A. Salama, M.M. (2008). One day ahead prediction of wind speed and direction", IEEE Transactions on Energy Conversion, 23, pp. 191 – 201.

Gandomkar, A. & Kiarasi, K. (2006). Wind energy potential estimation in Iran, 21st international power system conference.

Giebel, G. & Brownsword, R. & Kariniotakis, G. & Denhard, M. & C. Draxl, (2011). The state of the art in short term prediction of wind power a literature overview, 2nd ed. project report for the Anemos. Plus and safe wind projects. Roskilde, Denmark: Riso.

Gnana Sheela, K. & Deepa, S.N. (2013), neural network based hybrid computing model for wind speed prediction, Neurocomputing122, pp. 425-429, 2013.

Guo, Z.H. & Wu, J. & Lu, H.Y. & Wang, Z.J. (2011). A case study on hybrid wind speed forecasting method using BP neural network, Knowledge-Based Systems, 24, pp. 1048-1056, 2011.

Haykin, S. (1999). Dynamic driven recurrent networks in neural networks a comprehensive foundation, Prentice Hall, pp. 732-783.

Hu, J. & Wang, J. & Zeng, G. (2013).A hybrid forecasting approach applied to wind speed time series, Renewable Energy, 60, pp. 185-194.

Jongh, J.A. & Raghavan, K. & Van Huulle, F.J.L. (1996). Power for the world- A common concept, study, ECN-CE95e037.

Kalogirou, S.A. (2000). Applications of artificial neural-networks for energy systems, Applied Energy, 67, pp. 17-35.

Kalogirou, S.A. (2001). Artificial neural networks in renewable energy systems applications, Renewable and Sustainable Energy review, 5, pp. 373-401.

Kavasseri, R.G. & Seetharaman, K. (2009). Day-ahead wind speed forecasting using f-ARIMA models, Renewable Energy, 34, pp. 1388–1393.

Lei, M. & Shiyan, L. & Chuanwen, J. & Hongling, L. & Yan, Z. (2009). A review on the forecasting of wind speed and generated power, Renewable and Sustainable Energy Reviews, 13, pp. 915–920.

Liu, H. & Chen, C. & Tian, H. & Li, Y. (2012). A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks, Renewable Energy, 48, pp. 545-556.

Liu, H. & Tian, H. & Li, Y. (2012). Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction, Applied Energy, 98, pp. 415-424, 2012.

Liu, H. & Tian, H. & Chen, C. & Li, Y. (2013). An experimental investigation of two Wavelet-MLP hybrid frameworks for wind speed prediction using GA and PSO optimization, Electrical Power and Energy Systems, 52, pp. 161–173.

Liu, D. & Niu, D. & Wang, H. & Fan, L. (2014). Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm, Renewable Energy, 62, pp. 592-597.

Palau, A. & Velo, E. & Puigjanner, L. (1999). Use of neural networks and expert systems to control a gas/solid sorption chilling machine", International Journal of Refrigeration, 22, pp. 59–66.

Shi, J. & Guo, J. & Zheng, S. (2015). Evaluation of hybrid forecasting approaches for wind speed and power generation time series" Renewable and Sustainable Energy Reviews, 16, pp. 3471– 3480.

Smith, B.A. & McClendon, R.W. & Hoogenboom, G., (2007). Improving air temperature prediction with artificial neural networks, International Journal of Computer, Information, Systems and Control Engineering, 1, pp. 3155-3162.

Venayagamoorthy, G.K. & Rohrig, K. & Erlich, I. (2012). One Step Ahead: Short-term wind power forecasting and intelligent predictive control based on data analytics, IEEE Power and Energy Magazine, 10, Issue 5, pp 70-78.

Venkadesh, S. & Hoogenbooma, G. & Potter, W. & McClendon, R. (2012) "A genetic algorithm to refine input data selection for air temperature prediction using artificial neural networks" Applied Soft Computing, 13, pp. 2253–2260.

Yeh, W.C. & Yeh, Y.M. & Chang, P.C. & Ke, Y.C. & Chung, V. (2014). Forecasting wind power in the Mai Liao Wind Farm based on the multi-layer perceptron artificial neural network model with improved simplified swarm optimization, Electrical Power and Energy Systems, 55, pp. 741-748.

Mehrian, Nowruzpour, et al. "Dynamic Response for a Functionally Graded Rectangular Plate Subjected to Thermal Shock Based on LS Theory." Applied Mechanics and Materials. Vol. 332. Trans Tech Publications, 2013.

Vaziri, M. R., et al. "Modification of Shock Resistance for Cutting Tools Using Functionally Graded Concept in Multilayer Coating." Journal of Thermal Science and Engineering Applications 7.1 (2015): 011014.

Mehrian, Nowruzpour, Seyed Mohsen, and Mohammad Hasan Naei. "Two Dimensional Analysis of Functionally Graded Partial Annular Disk under Radial Thermal Shock Using Hybrid Fourier-Laplace Transform." Applied Mechanics and Materials. Vol. 436. Trans Tech Publications, 2013.

Mehrian, S. M., and S. Zamani Mehrian. "Modification of Space Truss Vibration Using Piezoelectric Actuator." Applied Mechanics and Materials. Vol. 811. Trans Tech Publications, 2015.

Mehrian, S. M., Amin Nazari, and Mohammad Hasan Naei. "Coupled Thermoelasticity Analysis of Annular Laminate Disk Using Laplace Transform and Galerkin Finite Element Method." Applied Mechanics and Materials. Vol. 656. Trans Tech Publications, 2014.

Mehrian, Sh Zamani, et al. "Structural health monitoring using optimising algorithms based on flexibility matrix approach and combination of natural frequencies and mode shapes." International Journal of Structural Engineering 7.4 (2016): 398-411.

Parsaei, M. R., Taheri, R., & Javidan, R. (2016). Perusing The Effect of Discretization of Data on Accuracy of Predicting Naïve Bayes Algorithm. Journal of Current Research in Science, (1), 457-462.

Parsaei, M. R., & Salehi, M. (2015, November). E-mail spam detection based on part of speech tagging. In 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) (pp. 1010-1013). IEEE.

Parsaei, M. R., Javidan, R., & Sobouti, M. J. (2016). Optimization of Fuzzy Rules for Online Fraud Detection with the Use of Developed Genetic Algorithm and Fuzzy Operators. Asian Journal of Information Technology, 15(11), 1856-1864.