A RECURRENT NEURAL NETWORK APPROACH FOR SHORT-TERM WIND POWER PREDICTION

ALBU RĂZVAN-D*, POPENȚIU-VLĂDICESCU F.*, KALMUȚCHI G.**
*University of Oradea, Universității no.1, Oradea,
**Technische Universität München, München, Germany
ralbu@uoradea.ro

Abstract - In this paper we propose a new method for solving the wind-power prediction problem, based on a recurrent neural network approach. Wind energy is free but the power supply generated from wind energy is not known in advance. Short-term wind power prediction becomes an extremely important field of research for the energy sector. Romania has a great potential for growth on short- to medium-term, for the wind energy industry. Because of its location along the western shore of the Black Sea, the average wind speed stands at about 25.2 kilometers per hour. The conferred results validate the proficiency of the described technique in short-term wind power forecast.

Keywords: wind-power, short-term, prediction, forecast, recurrent neural networks, wind energy.

1. INTRODUCTION

Romania has a great potential for growth on short- to medium-term, for the wind energy industry. Actually, Romania has the best wind energy sites in Eastern Europe according to Vestas Wind System A/S, the world's largest maker of wind turbines. Hans Joern Rieks, the company's president for central and Eastern Europe, said at a press conference in Bucharest, that Romania may produce as much as 14,000 megawatts of wind energy and may develop into a sustainable market. A study of Erste Bank considers Romania and especially the Region of Dobrogea with Constanta and Tulcea counties as the second best place in Europe to build WPP (Wind Power Plants). The average wind speed stands at about 25.2 kilometers per hour because of its location along the western shore of the Black Sea. The local industry has the potential to generate as much as 30.7 billion kilowatt-hours a year, powering the equivalent of Ireland, Serbia, or Peru and giving Romania an edge against other East European nations. Vestas has installed 22 wind turbines in Romania, with a total capacity of 44 megawatts, according to the latest data available as of June 30 2010. The Randers, Denmark-based Company currently has under construction three wind projects in Romania in the southern region of Dobrogea. Two of them, with a total capacity of 228 megawatts, are for EDP-Energias de Portugal SA, Portugal's biggest utility. Other companies installing turbines in Romania include EDP, CEZ AS, which is the Czech Republic's largest power distributor, E.ON, Germany's biggest utility, Iberdrola SA and Enel SA. Wind energy is free but the power supply generated from wind energy is not known in advance. Electricity generated from wind power can be highly variable at several different timescales: from hour to hour, daily, or seasonally. Annual variation even if exists it is not as significant. The short-term (hourly or daily) predictability of wind plant output is related to variability.

Wind energy, like other electricity sources, must be "scheduled". Wind power forecasting methods are used, but predictability of wind plant output remains low for short-term operation. Short-term wind power prediction becomes an extremely important field of research for the energy sector, since the system operators must handle an important amount of variable power.

In the technical literature, we can find two major approaches [1] to forecast wind power:

1. physical methods:
   - Require many physical considerations to gain the best prediction precision.
   - The input variables will be physical or meteorology information.
   - They present advantages in long-term prediction.
2. statistical methods:
   - Aspire at finding a relationship between the on-line measured power data.
   - They will use the historical data of the wind farm.
   - Do well in short-term prediction,
   - They are time-series-based models: auto regressive (AR) and auto regressive integrated moving average (ARIMA).

In the recent years, it has been reported that artificial-based models outperformed others in short-term prediction [2, 3].

This article is organized as follows: after a short discussion about recurrent neural networks where we argue why we chose them, we focus on learning with fixed points and in the third chapter, we describe the propose RNN with its architecture. The fourth chapter contains numerical results from a real world case study, particularly our RNN prediction results. We tested the proposed RNN using data sets collected from the ANM (National Meteorology Administration) website. In the last chapter, we raise some interesting conclusions and talk about plans and future works.

2. RECURRENT NEURAL NETWORKS

2.1. Why RNNs?

Before answering this question, we would like to show why we chose neural networks in the first place. Neural networks have been successfully applied for a long time in various domains such as research, business and industrial environments [4, 5]. They have been used to solve various problems like data prediction, classification and function approximation. Neural networks (NN) are simple, powerful and flexible tools for forecasting but they have some needs [6, 7, 8]. First, we must verify if there are enough data for training. An adequate selection of the input–output samples, an appropriated number of hidden units and enough computational resources available are other needs required by a NN based solution. NNs are data-driven and have the well-known advantages:

- They are able to approximate any nonlinear function.
- They can solve problems where the input–output relationship is neither well defined nor easily computable.
- Knowledge is automatically acquired during the learning process but this knowledge cannot be extracted from the trained network.

Many of the real-world problems, which one might think would require recurrent architectures, have solvable solutions by using layered architectures. For this reason, we would advise engineers to try, first, the layered architecture before resorting to the “big gun" of recurrence. The recurrent networks are often avoided because of the fear for unreasonable learning hours and incomprehensible algorithms and mathematics. Therefore, there is no reason to use a recurrent network when a layered architecture suffices. On the other hand, if recurrence is needed, there is the availability of an overabundance of learning algorithms. The reason for exploring recurrent architecture lies in their potential for dealing with temporal behaviors.

However, the question still rises: “Why RNNs for short-term wind power prediction when we already have so much layered architecture that could solve this problem?” The relative superiority of recurrent networks to feed-forward networks in forecasting is not just due to its ability to model time series data with lower errors, but rather to model a parsimonious training set. With the rapid growth of processing speed in the last years, the context in which we define an efficient method changed. The capabilities of many digital electronic devices are strongly linked to Moore's law. Processing speed, memory capacity, sensors and even the number and size of pixels in digital cameras are improving at exponential rates. This exponential improvement has given us a new perspective on the prediction problem. Nowadays the learning time, a parameter that was critical in designing an efficient NN based solution, is not such a big problem like 10 years ago, because we already have on the market processors similar to Intel Core i7 990x that can reach speeds of 4.5-5 GHz. Consequently, we propose a model for the problem of short-term wind power prediction focused more on the prediction accuracy then the learning time.

2.2. Learning with fixed points

One problem with fixed points is that recurrent networks do not always converge to them [9, 10]. There are a number of special cases that guarantee convergence to a fixed point. Some simple linear conditions on the weights such as zero-diagonal symmetry ($w_{ij}=w_{ji}$, $w_{ii}=0$) guarantee that Lyapunov function decreases until a fixed point is obtained:

$$L = \sum_i (y_i log y_i + (1 - y_i)\log(1 - y_i)) - \sum_{ij} w_{ij}\, y_i y_j \quad (1)$$

W_{ij} are the weights of the connections from unit i to unit j and y_i is the activation level of unit i:

$$x_i = \sum_j w_j\, y_j \quad (2)$$

In addition, the general equation that model the neural network is:

$$\frac{dy_i}{dt} = I_i + \emptyset(x_i) - y_i \quad (3)$$

Where Ǿ is an arbitrary differentiable function and I_i are the inputs. Aliya has shown that a unique fixed point would be achieved regardless the initial conditions, if:

$$\tau > \max(\acute{\emptyset}). \quad (4)$$

Other empirical studies show that applying fixed point learning algorithms stabilizes the networks [11, 12]. However, the fixed point learning algorithms can still have problems even when it is guaranteed that a network settles to a fixed point. The learning procedures compute the derivative of some error measure. This gradient is used by an optimization procedure in order to minimize the errors. The optimization procedures assume that the mapping from the network's internal parameters to the resulting errors is continuous and can fail when this assumption is violated. This means that the learning algorithm changes the locations of the fixed points by varying the weights. Accordingly, it is also possible for a result to stumble upon such a discontinuity. This will induce errors, which will appear suddenly.

3. THE PROPOSED METHOD

The time scales we use in this short-term prediction solution are in the order of some days for the forecast horizon and from minutes to hours for the time-step. For the purpose of time series prediction, a RNN can be considered to be a general nonlinear mapping between a subset of the past time series and the future time series values. The proposed architecture is presented in Fig. 1.

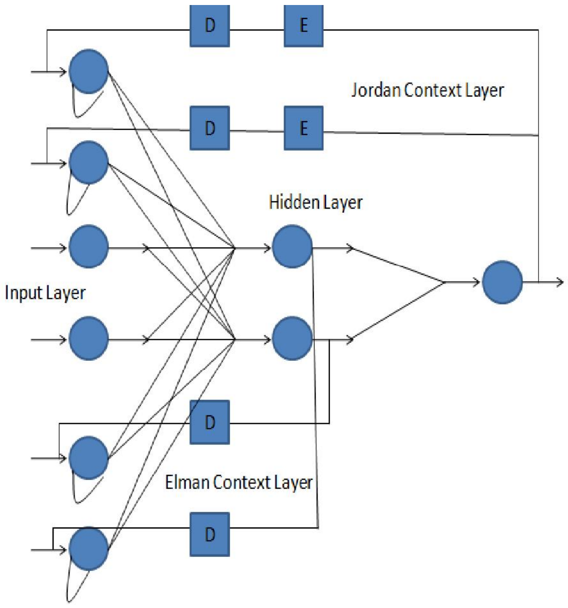


Figure 1 The proposed RNN

The RNN has two context layers: the Elman context layer and the Jordan context layer, both with some differences from the original Elman and Jordan recurrent neural networks. The Elman context layer differs from the original Elman RNN because the two context neurons obtain inputs from the output of the hidden layer after a delay of one time unit, and from itself. In the Jordan context layer the difference is that the context neurons obtain inputs from the output error of the network after a delay of one time unit, and from itself. In both context layers there are two neurons with self-feedbacks. For predicting time series in the output layer, we need just one neuron. We use also two neurons in the input layer because it has been reported that every data point in a time series is only strongly dependent on the immediate past two values [12, 13, 14]. The linear activation function is used in the output layer, the Jordan context layer and the Elman context layer. The sigmoid activation function is used in the hidden layer.

4. A CASE STUDY

4.1. Neuro Solutions

Neuro Solutions is a graphical neural network development tool, which can easily create a neural network model for the input data. This software combines a modular design interface with advanced learning procedures. Neuro Solutions provides the power and flexibility needed to design the best neural network for our problem.

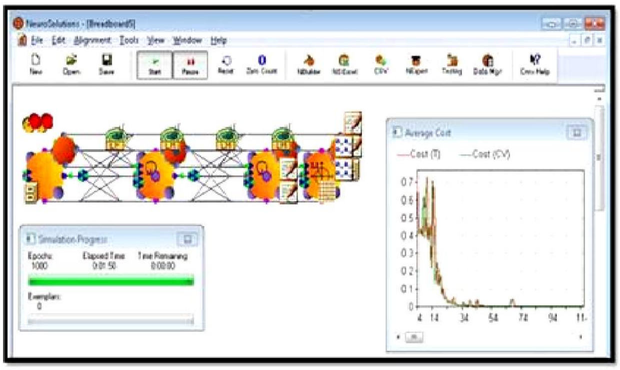


Figure 2 Neuro Solution Breadboard

4.2. Calculation of wind power

Wind is made up of moving air molecules [15] and these have mass. Any moving object with mass carries kinetic energy in an amount, which is given by the equation:

$$\text{Kinetic Energy} = 0.5 \text{ x Mass x Velocity}^2 \qquad (5)$$

Air has a known density (around 1.23 kg/m^3 at sea level), so the mass of air hitting our wind turbine (which sweeps a known area) each second is given by the following equation:

$$\text{Mass/sec (kg/s)} = \text{Velocity (m/s) x Area (m}^2\text{) x Density (kg/m}^3\text{)} \qquad (6)$$

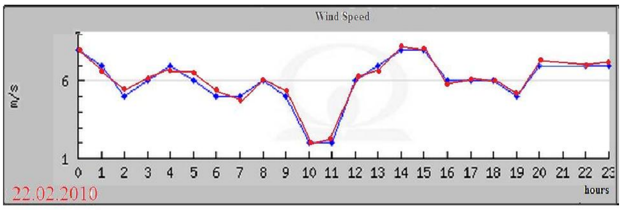
Thus, the power (i.e. energy per second) in the wind hitting a wind turbine with a certain swept area is given by simply inserting the mass per second calculation into the standard kinetic energy equation given above resulting in the following vital equation:

$$\text{Power} = 0.5 \text{ x Swept Area x Air Density x Velocity}^3 \qquad (7)$$

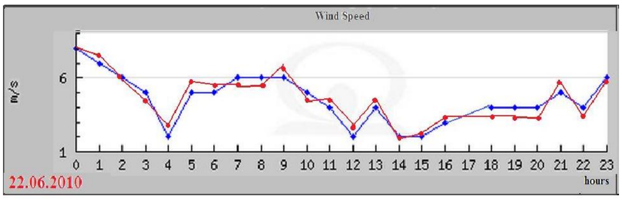
Wind speeds in most of the world can be modeled using the Weibull Distribution [16]. This statistical tool tells us how often winds of different speeds will be seen at a location with a certain average wind speed. Knowing this helps us to choose a wind turbine with the optimal cut-in and cut-out speeds [17]. The cut-in speed is the wind speed at which the turbine starts to generate usable power. The cut-out speed is the speed at which the turbine hits the limit of its alternator and can no longer put out increased power output with further increases in wind speed.

4.3. Prediction results

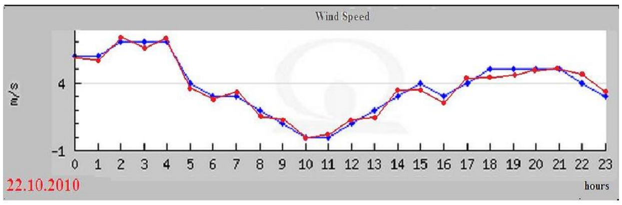
We tested the proposed RNN using data sets collected from the ANM (National Meteorology Administration) website. The prediction results are shown below:



a)



b)



c)

Figure 3 Prediction results a) a winter day b) a summer day c) a fallen day

We randomly selected three days, one from each season, and predict the wind speed at each hour for these days. Our RNN results are illustrated in figure above. The blue line represents the real measured wind speeds collected from the ANM web site and the red line indicates the predicted values. To predict W (d, h), the wind speed at hour h of day d, we train the RNN with only last two values: W (d-1, h), W (d-2, h); it has been reported that every data point in a time series is only strongly dependent on the immediate past two values. The training is complete when we provide as inputs all wind speed values, for a number of n epochs. One epoch is finished when the entire training set is exposed to the RNN. The number of epochs is the number of steps of the training process, it is a dynamic value; we set it high and let it stop according to the validation set. The initial learning rate is 0.001, results in good coarse training quickly. For better performance, we used a schedule of 0.0005 for two epochs, followed by 0.0002 for the next three, 0.0001 for the next three, 0.00005 for the next four, and 0.00001 thereafter. The learning rate is decreased by 79.4% of its value after every epoch. In order to implant fixed points into recurrent systems, the backpropagation technique is used. In fixed-point learning, the first action is the forward propagation of the activations. This procedure repeated for a certain number of times will induce the relaxation period. This has to be repeated until the network attains its own dynamic. After the net become stable, an error can be computed at the output. Then, the error is propagated backwards through the network. The error at each output can be multiplied by the relaxed activation for updating the weights. We have to select the relaxation time both in the forward and backpropagation phases.

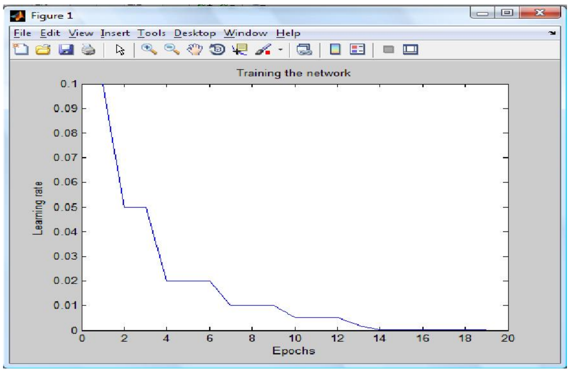


Figure 4 Learning rate evolution by epochs

Table 1. Prediction errors

Data/hour	Winter day 22.02.2010	Summer day 22.06.2010	Fallen day 22.10.2010
0	+0.11	+0.12	-0.01
1	-0.21	+0.3	-0.1
2	+0.3	-0.1	+0.1
3	+0.13	-0.4	-0.2
4	-0.21	+0.3	+0.1
5	+0.3	+0.3	-0.1
6	+0.22	+0.2	-0.02
7	-0.2	-0.2	+0.1
8	+0.01	-0.2	-0.2
9	+0.12	+0.2	+0.17
10	-0.01	-0.3	-0.02
11	+0.13	+0.2	+0.12
12	-0.02	+0.1	+0.21
13	-0.21	+0.15	-0.14
14	+0.2	-0.02	+0.13
15	+0.11	-0.01	-0.23
16	-0.2	+0.1	-0.11
17	-0.12	+0.01	+0.09
18	-0.03	-0.2	-0.21
19	-0.01	-0.2	-0.2
20	+0.22	-0.26	-0.02
21	+0.12	+0.2	+0.01
22	+0.03	-0.2	+0.19
23	+0.01	-0.1	+0.13
AVERAGE	+0.032	-0.002	-0.008
AVERAGE OF ABSOLUTE VALUES	0.134	0.182	0.121

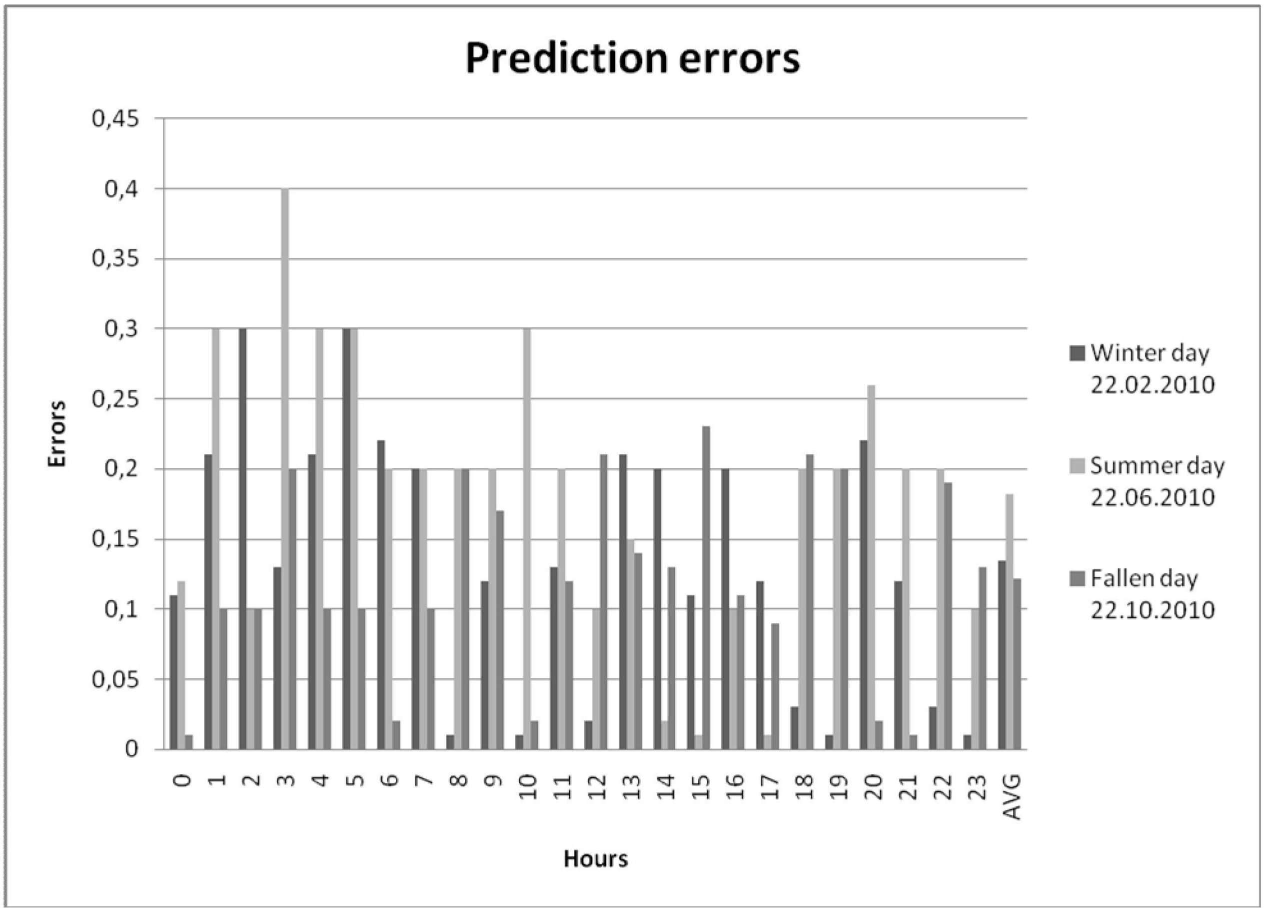


Figure 5 **Prediction errors for each day and average values**

As we can see in the figure 5, we obtained very good prediction results proven by a very low average error rate. The secret is the joint usage of Neuro Solutions features and our innovative RNN architecture. Using Neuro Solutions 6, we can select the Fixed Point radio button in the Dynamic Control Inspector. If the network is not relaxed enough, the output activation will not be in the steady state and will produce an erroneous error estimate. The transmitters are a class of objects that test for a particular condition and perform global communications within Neuro Solutions breadboard. Transmitters have many potential applications, but here they will be used for controlling the relaxation time of the network in the forward and backward plans. The relaxation can be controlled by measuring the differential between two consecutive iterations. When the difference is smaller than a given threshold, we can assume that the network is stable.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a RNN prediction model. This model is developed for the short-term wind power prediction based on the data collected from the ANM (National Meteorology Authority) web site. The presented results validate the proficiency of the proposed approach in short-term wind power prediction. Higher value of neurons in hidden layer may force the network to memorize. Lower value of neurons in hidden layer, would waste a great deal of training time in finding its optimal representation. More neurons require more computation, but they allow the network to solve problems that are more complicated. We plan to optimize the learning process to perform faster and develop a software solution based on the proposed method:

ACKNOWLEDGEMENTS

This work was partially supported by the strategic grant POSDRU/CPP107/DMI1.5/S/80272, Project ID80272 (2010), co-financed by the European Social Fund-Investing in People, within the Sectorial Operational Program Human Resources Development 2007-2013

REFERENCES

[1]. Catalão JPS, Pousinho HMI, Mendes VMF. An artificial neural network approach for short-term wind power forecasting in Portugal. Eng Intell Syst Electr Eng Commun 2009;17(1):5–11.

[2]. Charles Thompson, Fighting Virus and Hack Attacks with a Network Analyzer, IEEE transactions on reliability, VOL. 59, NO. 1, 2010.

[3]. Delarue ED, Luickx PJ, D'haeseleer WD. The actual effect of wind power on overall electricity generation

costs and CO2 emissions. Energy Convers Manage 2009;50(6):1450–6.

[4]. Eduardo O.C., Ramirez Pozo, A.T., & Vergilio, S.R., A Genetic Programming Approach for Software Reliability Modeling, IEEE transactions on reliability, VOL. 59, NO1, 2010.

[5]. Fan S, Liao JR, Yokoyama R, Chen LN, Lee WJ. Forecasting the wind generation using a two-stage network based on meteorological information. IEEE Trans Energy Convers 2009;24(2):474–82.

[6]. Gebraeel N., & Pan, J., Prognostic degradation models for computing and updating residual life distributions in a time-varying environment, IEEE Trans. Reliability, vol. 57, no. 4, pp. 539–550, 2009.

[7]. Gebraeel, N., Elwany A., & Pan, J., Residual life predictions in the absence of prior degradation knowledge, IEEE Trans. Reliability, vol. 58, no. 1, pp. 106–117, 2009.

[8]. Hocaoglu FI, Fidan M, Gerek ON. Mycielski approach for wind speed prediction. Energy Convers Manage 2009;50(6):1436–43.

[9]. Melício R, Mendes VMF, Catalão JPS. Fractional-order control and simulation of wind energy systems with PMSG/full-power converter topology. Energy Convers Manage 2010;51(6):1250–8.

[10]. Müller, A., & Suhner, M.-C. Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system, Reliability Engineering and System Safety, vol. 93, no. 2, pp. 234–253, 2008.

[11]. Peysson, F., & Ouladsine, M. A generic prognostic methodology using damage trajectory models, IEEE Trans. Reliability, vol. 58, no. 2, pp. 277–285, 2009.

[12]. Peysson, F., Ouladsine, M., Outbib, R., Leger, J.-B., Olivier Myx, O., & Allemand, C., A Generic Prognostic Methodolog Using Damage Trajectory Models, IEEE transactions on reliability, VOL. 58, NO. 2, 2009.

[13]. Ramirez-Rosado IJ, Fernandez-Jimenez LA, Monteiro C, Sousa J, Bessa R. Comparison of two new short-term wind-power forecasting systems. Renew Energy 2009;34(7):1848–54.

[14]. Shoorehdeli MA, Teshnehlab M, Sedigh AK, Khanesar MA. Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods. Appl Soft Comput 2009;9(2):833–50.

[15]. Tian Z., & Ming, J.Z., Health Condition Prediction of Gears Using a Recurrent Neural Network Approach, IEEE transactions on reliability, VOL. 59, NO. 4, 2010.

[16]. Yun Z, Quan Z, Caixin S, Shaolan L, Yuming L, Yang S. RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment. IEEE Trans Power Syst 2008;23(3):853–8.

[17]. Zubia I, Ostolaza X, Susperregui A, Tapia G. Complete wind farm electromagnetic transient modeling for grid integration studies. Energy Convers Manage 2009;50(3):600–10.