

## Research Article

# Developing a Local Neurofuzzy Model for Short-Term Wind Power Forecasting

E. Faghihnia,<sup>1</sup> S. Salahshour,<sup>1</sup> A. Ahmadian,<sup>2</sup> and N. Senu<sup>2</sup>

<sup>1</sup> Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

<sup>2</sup> Mathematics Department, Science Faculty and Institute for Mathematical Research, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia

Correspondence should be addressed to E. Faghihnia; faghihnia.elahe@gmail.com

Received 24 April 2014; Revised 2 June 2014; Accepted 16 June 2014; Published 16 July 2014

Academic Editor: Hossein Jafari

Copyright © 2014 E. Faghihnia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Large scale integration of wind generation capacity into power systems introduces operational challenges due to wind power uncertainty and variability. Therefore, accurate wind power forecast is important for reliable and economic operation of the power systems. Complexities and nonlinearities exhibited by wind power time series necessitate use of elaborative and sophisticated approaches for wind power forecasting. In this paper, a local neurofuzzy (LNF) approach, trained by the polynomial model tree (POLYMOT) learning algorithm, is proposed for short-term wind power forecasting. The LNF approach is constructed based on the contribution of local polynomial models which can efficiently model wind power generation. Data from Sotavento wind farm in Spain was used to validate the proposed LNF approach. Comparison between performance of the proposed approach and several recently published approaches illustrates capability of the LNF model for accurate wind power forecasting.

## 1. Introduction

Being free and environmentally friendly, the wind energy is utilized growingly as a renewable source of energy. According to the World Energy Outlook 2010 [1] published by International Energy Agency (IEA), worldwide wind generation capacity is projected to increase to over 1000 GW by 2035 which shows a dramatic growth compared to 120 GW in 2008. It is worth noting that world wind generation increased by about 20% from 2008 to 2009 and has more than tripled since 2004 [2].

Despite the favourable characteristics of wind power, integration of the large scale wind generation capacity into power systems is challenging. This is due to the variability and uncertainty of the wind power, resulting from variable nature of earth atmosphere [3]. Conventional unit commitment procedure is not capable of time-ahead scheduling of the system in the presence of wind uncertainty since wind-generated power is not known in advance [4]. Furthermore, higher level of spinning reserve is required to deal with wind uncertainty and maintain security of the power system [5].

For the aforementioned reasons, accurate and reliable information about the future values of wind power generation is of utmost importance. During the past decade, many methods have been developed for the wind power and speed forecasting. Generally, these approaches can be classified into two broad categories, namely, physical methods and time series methods [6]. Physical methods use physical and meteorological information, including description of orography, roughness, obstacles, pressure, and temperature to model wind power and forecast its future values. These approaches perform satisfactorily for long-term prediction of wind power [7].

On the other hand, time series approaches require a smaller volume of data and information, compared to physical methods. Some of key meteorological variables such as wind speed and direction are needed by a time series approach to build the wind forecast model. Sometimes, only the historical data of the generated wind (or Wind) power is used by the time series models to forecast the wind power [8]. Conventional statistical models such as autoregressive (AR) models, autoregressive integrated moving average (ARIMA)

models, and GARCH models have been proposed for wind speed and power forecasting [6, 9].

Computational-intelligence- (CI-) based approaches have been also used for wind power forecasting. It has been reported that CI-based approaches can outperform physical and conventional time series models in short-term wind power forecasting. For instance, combination of artificial neural networks and ARIMA models has been proposed for wind power forecasting in Mexico [10]. Amjady et al. proposed a hybrid neural network model for short-term wind power forecasting [8]. They used particle swarm optimization (PSO) algorithm for optimizing the structure of their proposed neural network and applied their approach to wind power forecasting in Alberta, Canada, and Oklahoma, USA. In another study, Amjady et al. [5] developed a ridgelet neural network (RNN) for day-ahead wind forecasting. Although they reported acceptable forecasting results, using fixed weights in the structure of RNN limits the performance of the network. ANFIS model has also been used by Catalão et al. [7] for day-ahead wind power forecasting. But, the procedure of training ANFIS model and fine tuning of its parameters are very complicated and time consuming, especially when the number of input variables is large. Borg and Rothkrantz [11] proposed a radial basis networks method to predict aggregate wind power production from a number of wind farms. Li et al. [12] presented a two-step methodology for wind speed forecasting based on Bayesian combination algorithm and three neural network models, namely, adaptive linear element network (ADALINE), backpropagation (BP) network, and radial basis function (RBF) network. Louka et al. [13] applied a Kalman filter as a postprocessing method in numerical forecast of wind speed in order to eliminate systematic errors in the data. Using a generic algorithm (GA) approach for optimizing a fuzzy inference system (FIS) model is reported to results in some degree of accuracy [14]. For a detailed review of wind power forecasting models, refer to [4, 15, 16].

In this paper, the elaborated local neurofuzzy (LNF) model is proposed for short-term wind power forecasting. Powered by the learning ability of the neural networks and incorporation of the a priori knowledge and transparency in the fuzzy systems, the LNF models can efficiently describe the complex processes and systems. The proposed LNF model is established based on a *divide-and-conquer* strategy, trying to solve a complicated problem (e.g., forecasting of nonlinear wind power series) by breaking it down to a number of smaller, and thus simpler, subproblems. Furthermore, as opposed to [5], employing higher degree polynomials (which are flexible and powerful in modelling nonlinear processes) as local models in the structure of LNF model allows for modelling and description of highly nonlinear and complex behaviour of wind power series.

The mathematical description of the LNF model and the corresponding polynomial model tree (POLYMOT) learning algorithm are presented in Section 2 and Section 3, respectively. In Section 4, the proposed LNF approach is applied to forecasting wind power for Sotavento wind farm, Spain, in four different months of 2010. Finally, a concluding summary is given in Section 5.

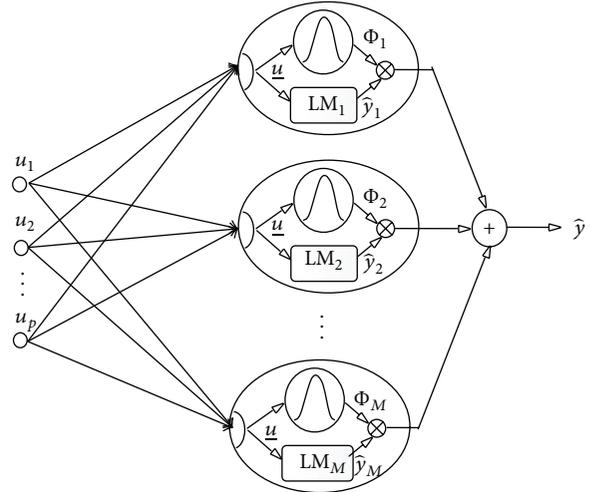


FIGURE 1: Structure of LNF with  $p$  inputs and  $M$  local models.

## 2. Local Neurofuzzy Models

Neurofuzzy (NF) models are fuzzy models that are not solely designed by expert knowledge but are at least partly learned from data [17]. In fact, an NF model is a fuzzy system drawn in a neural network structure and thus learning methods already developed for neural networks can be applied to the NF model. Therefore, the neurofuzzy systems inherit the learning capability of the neural networks as well as the logicity and transparency in the fuzzy systems [17]. Local neurofuzzy models are an appealing class of neurofuzzy systems and work based on the interpolation of the local models [18]. In the LNF approach, the whole input space is partitioned into a set of subregions, each is determined by its corresponding validity function and local model (LM). Interestingly, the procedure of input space partitioning allows describing a complex nonlinear process by creating a number of simpler local models, whose parameters are easily identifiable.

The general mathematical expression for an LNF with  $p$ -dimensional input,  $\underline{u} = [u_1, u_2, \dots, u_p]^T$ , and  $M$  local models is given by

$$\hat{y} = \sum_{i=1}^M h_i(\underline{u}) \Phi_i(\underline{u}), \quad (1)$$

where  $h_i(\cdot)$  is a nonlinear function describing  $i$ th local model ( $LM_i$ ),  $\Phi_i$  is the corresponding validity function of the  $LM_i$ , and  $\hat{y}$  is the LNF model's output. In order to have a smooth transition between local models, the validity functions are smooth and take their values between 0 and 1. Furthermore, the validity functions must form a partition of unity to have reasonable interpretation of local models:

$$\sum_{i=1}^M \Phi_i(\underline{u}) = 1. \quad (2)$$

The architecture of the LNF model, described by (1), is illustrated in Figure 1. As depicted in Figure 1, the total output

of the LNF model can be represented based on the output of each local model:

$$\hat{y} = \sum_{i=1}^M \hat{y}_i \Phi_i(\underline{u}), \quad (3)$$

where  $\hat{y}_i$  is the output of  $LM_i$  and is equal to  $h_i(\underline{u})$ .

It is worth noting that the arbitrary nonlinear functions  $h_i(\cdot)$ , and as a result arbitrary local models, can be utilized in the LNF model structure. This outstanding feature allows choosing complex local models in order to better model and describe complex and highly nonlinear processes and systems. In this paper, the focus is on polynomial functions and developing local polynomial models with arbitrary degree. The polynomial functions are powerful in describing nonlinear processes since polynomials with arbitrary degrees can be seen as Taylor series expansion of any unknown function. The next subsections describe the procedure of identification local models and validity functions.

**2.1. Identification of Local Models.** Considering a local polynomial model with degree  $n$ , the LNF model in (1) can be restated as

$$\hat{y} = \sum_{i=1}^M \left( \theta_{i,0} + \theta_{i,1}u_1 + \theta_{i,2}u_1^2 + \theta_{i,3}u_2 + \theta_{i,4}u_1u_2 + \theta_{i,5}u_2^2 + \dots + \theta_{i,l}u_p^n \right) \Phi_i(\underline{u}), \quad (4)$$

where  $\underline{\Theta}_i = [\theta_{i,0}, \theta_{i,1}, \dots, \theta_{i,l}]$  is the parameter vector of the local polynomial model  $i$ . For an  $n$  order polynomial function and with  $p$ -dimensional input space, the number of parameters of each local polynomial model will be

$$l = \frac{(p+n)!}{p!n!}. \quad (5)$$

For an efficient estimation of local model parameters, weighted least square algorithm is employed. In addition, it was assumed that the validity functions are known and predetermined. The weighted least square estimation was carried out based on the minimization of local error of each local polynomial model for target output samples. Consider

$$\min_{\underline{\Theta}_i} \left\{ I_i = \sum_{j=1}^N \Phi_i(\underline{u}(j)) e(j) \right\}, \quad (6)$$

where  $e(j) = y(j) - \hat{y}(j)$  and  $\underline{y} = [y(1), \dots, y(N)]$  are  $N$  target outputs.

Considering a second degree polynomial ( $l = 2$ ), the corresponding regression matrix is given by

$$\underline{X}_i = \begin{bmatrix} 1 & u_1(1) & \dots & u_p^2(1) \\ 1 & u_1(2) & \dots & u_p^2(2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_1(N) & \dots & u_p^2(N) \end{bmatrix}, \quad i = 1, \dots, M. \quad (7)$$

Given the regression matrix in (7), the solution of the weighted least square problem in (6) can be expressed as

$$\underline{\Theta}_i = \left( \underline{X}_i^T \underline{Q}_i \underline{X}_i \right)^{-1} \underline{X}_i^T \underline{Q}_i \underline{y}, \quad (8)$$

where  $\underline{Q}_i$  is the  $N \times N$  diagonal weighting matrix

$$\underline{Q}_i = \begin{bmatrix} \Phi_i(\underline{u}(1)) & 0 & \dots & 0 \\ 0 & \Phi_i(\underline{u}(2)) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \Phi_i(\underline{u}(N)) \end{bmatrix}, \quad (9)$$

$i = 1, \dots, M.$

Hence, the parameters of the local polynomial functions can be efficiently estimated using (8).

**2.2. Identification of Validity Functions.** Multivariate Gaussian functions are normally chosen as validity functions in the LNF model. The multivariate  $p$ -dimensional Gaussian function for the  $i$ th local model can be expressed as

$$\begin{aligned} \mu_i(\underline{u}) &= \exp \left( -\frac{1}{2} \left( \frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2} + \dots + \frac{(u_p - c_{ip})^2}{\sigma_{ip}^2} \right) \right) \\ &= \exp \left( -\frac{1}{2} \frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2} \right) \dots \exp \left( -\frac{1}{2} \frac{(u_p - c_{ip})^2}{\sigma_{ip}^2} \right), \end{aligned} \quad (10)$$

$i = 1, \dots, M,$

where  $\underline{c}_i = [c_{i1}, \dots, c_{ip}]$  and  $\underline{\sigma}_i = [\sigma_{i1}, \dots, \sigma_{ip}]$  represent centre coordinates and standard deviations of the Gaussian function. The Gaussian functions in (10) need to be normalized to form a partition of unity. Consider

$$\Phi_i(\underline{u}) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^M \mu_j(\underline{u})}, \quad i = 1, \dots, M. \quad (11)$$

The  $\underline{c}_i$  and  $\underline{\sigma}_i$  are the parameters of the Gaussian validity functions which should be estimated from the observation data. These consequent parameters are estimated using polynomial model tree (POLYMOT) algorithm, described in the next section. In the POLYMPT learning algorithm, the first step is to determine the parameters of the validity functions using a heuristic approach. Having known the validity functions, parameters of the local models are estimated using weighted least square algorithm, presented in Section 2.1. The POLYMOT learning algorithm increases the complexity of the model until desired performance is achieved.

### 3. Polynomial Model Tree (POLYMOT) Learning Algorithm

The POLYMOT learning algorithm belongs to the category of incremental tree construction algorithms [18]. The POLYMOT algorithm is, in fact, the modified version of

the local linear model tree (LOLIMOT) algorithm, which partitions the input space by axis-orthogonal splits into hyper-rectangles [17]. In the POLYMOT algorithm, in each iteration, a new local model is added to the LNF network or the number of parameters of the worst local model is increased (i.e., the degree of the worst local polynomial model is incremented). Then, the validity functions which correspond to the actual partitioning of the input space are computed and the parameters of the corresponding local polynomial models are optimized by the weighted least square technique. The POLYMOT algorithm can be summarized in the five steps, as stated below.

*Step 0* (start from an initial model). Set  $M = 1$  and start with a single first-order local model whose validity function  $(\Phi_1(\underline{u}))$  covers the whole input space.

*Step 1* (find the worst performing local model). Calculate the loss function  $I_i$  defined in (6), for all local models and find the worst performing LM.

*Step 2* (fit a higher degree polynomial). If increasing the polynomial order of the worst performing LM results in lower global model error, then increase the local model order and proceed to Step 4; otherwise, go to Step 3.

*Step 3* (split the input space). If increasing of the order of worst local model does not lower global model error, then

- (a) division of the worst LM into two equal halves must be tried in all  $p$ -dimensions. For each of  $p$  divisions, a multidimensional validity function must be constructed for both newly generated hyperrectangles. Gaussian membership functions are placed at the centres of the hyperrectangles and standard deviations are selected proportional to the extension of hyperrectangles (usually  $1/3$  of hyperrectangle's extension). Then, the rule consequent parameters of both new LMs must be estimated using weighted least square approach and finally the loss function for the current overall model must be computed.
- (b) The best LM related to the lowest loss function value must be selected and the number of local models is incremented:  $M \rightarrow M + 1$ .

*Step 4* (check termination criterion). If the termination criterion, for example, a desired level of model's performance or complexity, is met, then stop; otherwise, go to Step 1.

The POLYMOT learning algorithm yields maximum generalization and noteworthy forecasting performance of the identified LNF model. The presented algorithm is illustrated by a flowchart in Figure 2. For a better understanding of the POLYMOT algorithm, a three-dimensional graphical representation for execution of POLYMOT algorithm in a two-dimensional input space and up to the first four iterations is shown in Figure 3. In this figure, the input space is split into two halves in the first iteration. Then, in the second iteration, the order of second LM (2-2) is increased by one. Finally, LMM 2-2 is vertically divided into to new local models.

It should be noted that the maximum order of polynomial functions was limited to 3 in order to maintain the number of parameters of the local models to a reasonable level.

#### 4. Selection of Input Variables

Proper selection of forecast model's inputs substantially affects model's performance and accuracy of forecasting. Due to nonlinear nature of wind power series, a sophisticated method is required which can efficiently capture the nonlinear relevance between different wind power lags as well as the relevance between wind lags of wind power series and those of important exogenous variables, for example, wind speed and direction. In this paper, we propose utilization of mutual information (MI) input selection technique. Mutual information, as a measure of dependencies, is very powerful in assessing the relevance or redundancy of the input variables [19]. The concept of MI, which was originated from Shannon entropy, addresses the dependencies between random variables [20]. The MI between two random variables  $X$  and  $Y$ , denoted by  $I(X, Y)$ , expresses the amount of information shared by the variables. In other words, the MI between  $X$  and  $Y$  measures the reduction in uncertainty on  $X$  due to the knowledge provided by  $Y$  and vice versa. We skip the theoretical details of MI and proceed to input selection algorithm based on MI technique. More information about MI theoretical details can be found in a research developed by the authors in [19].

Selecting a set of input variables with the highest relevance to the output and the least interdependence among each other is the goal of the input selection algorithms. For this purpose, we will try to find and sort input variables which have large MI with the output variable and small MI with all other already selected input variables [19]. This MI-based input selection algorithm is illustrated by Figure 4. In fact, the illustrated algorithm in Figure 4 sorts the most relevant input features to the output based on the MI criterion, in a descending order.

#### 5. Wind Power Forecasting Results and Discussion

*5.1. Data Description, Input Selection, and Error Measures.* The wind power generation in Sotavento wind farm [21], located in Spain, in four different months of 2010 was forecasted using the proposed LNF approach. Spain is one the world's leading countries in utilization of wind power. In 2009, Spain contributed to 13.8% of global wind generation. It also produced over 12% of its electricity from wind generation in 2009 [2]. The Sotavento wind farm, which has been selected as our case study, includes 24 wind turbines and its nominal power is 17.56 MW [21]. The data used to construct forecast model and perform predictions include hourly wind power, wind speed, and wind direction. These data were collected from the website of the Sotavento wind farm [21], containing real-time and historical data.

The wind power generation in four different months of April, May, June, and July in 2010 was forecasted. It should

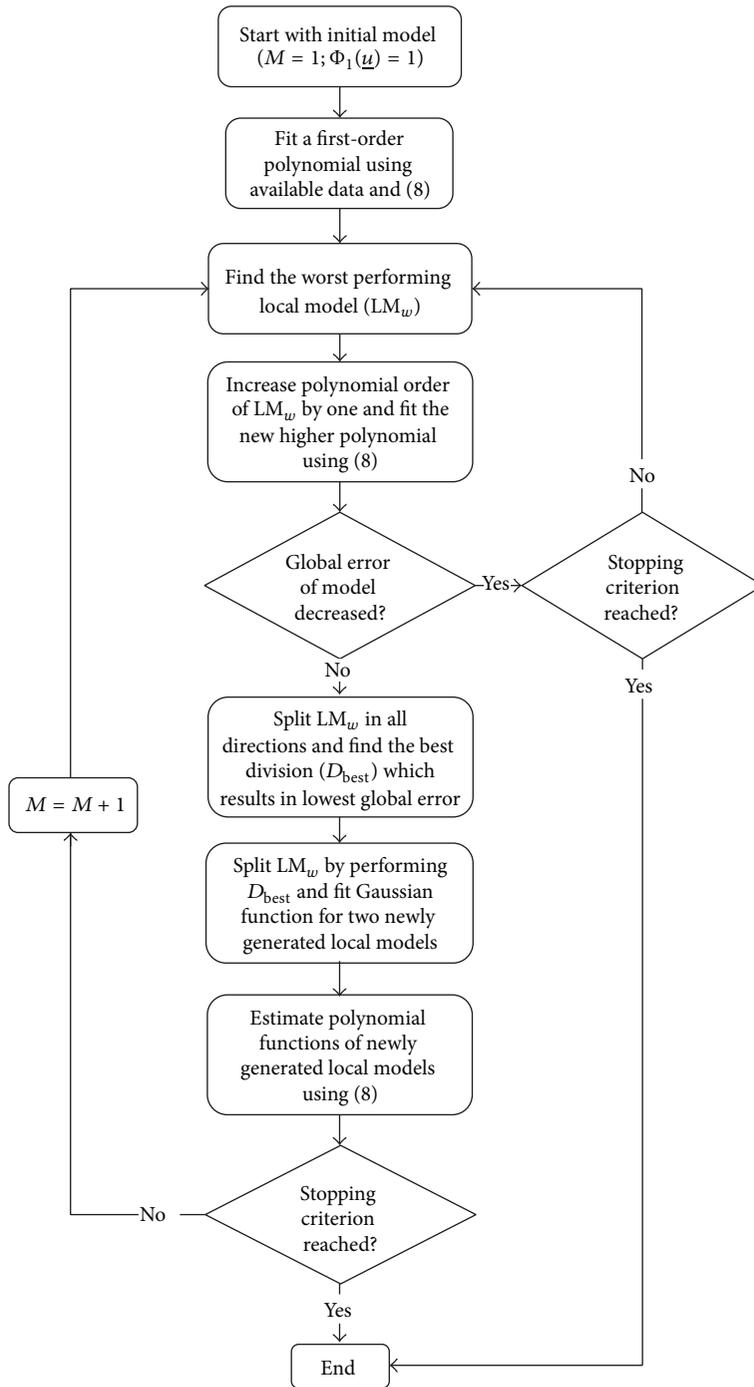


FIGURE 2: Flowchart of POLYMOT learning algorithm.

be noted that the forecasts are 24 steps ahead (one day ahead). That is, first the wind power generations in the first 24 hours of a test month are forecasted. Then, the forecasting window moves 24 steps ahead and the wind power generations for the second day (hours 25 to 48) are forecasted. This process continues on until the wind power generation for the whole test month was forecasted. The date information for training and test data for each test month are summarized in Table 1.

TABLE 1: Training and test data sets.

Training periods for the first test day	Test period
11 February–31 March 2010	April 2010
13 March–30 April 2010	May 2010
12 April–31 May 2010	June 2010
12 May–30 June 2010	July 2010

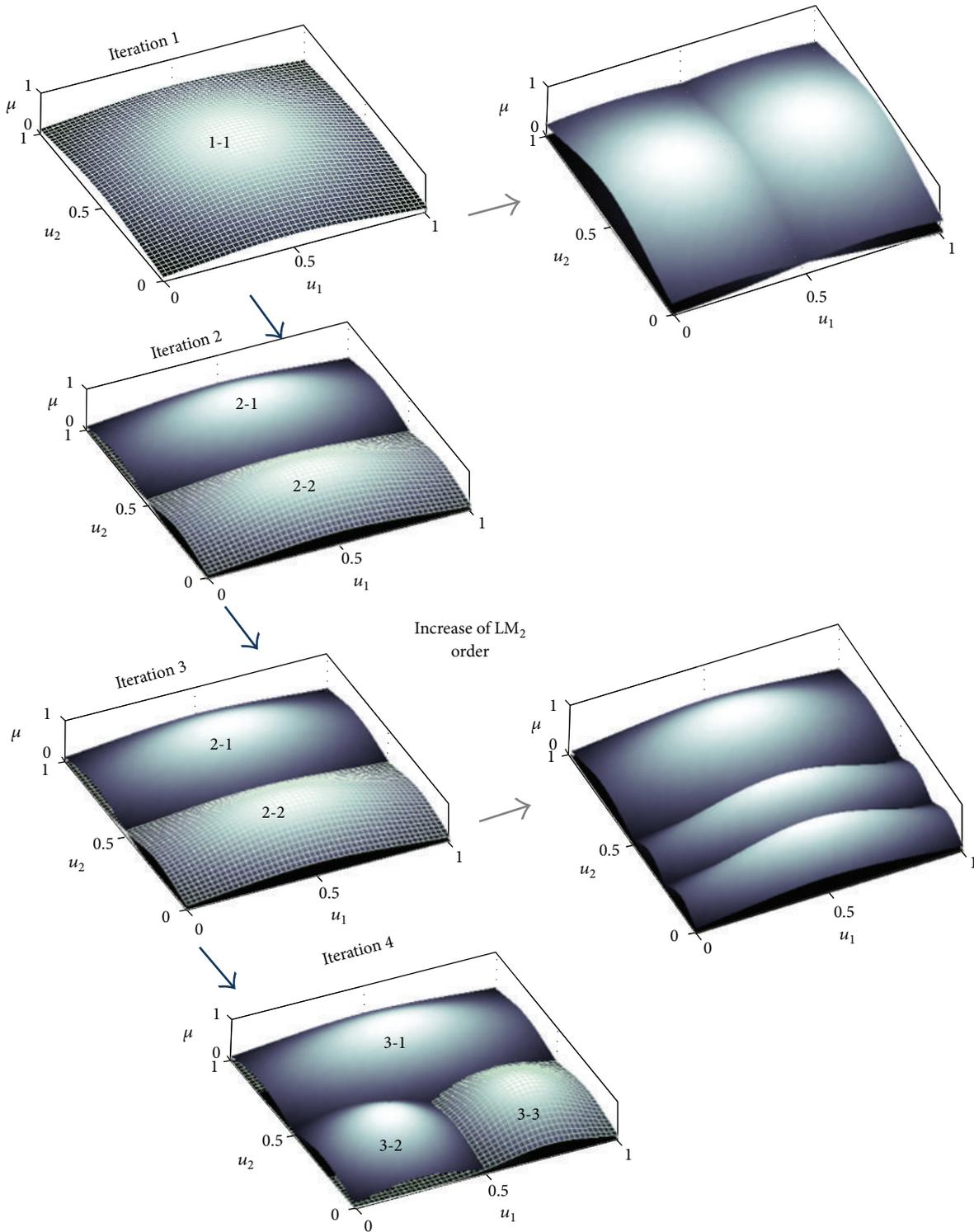
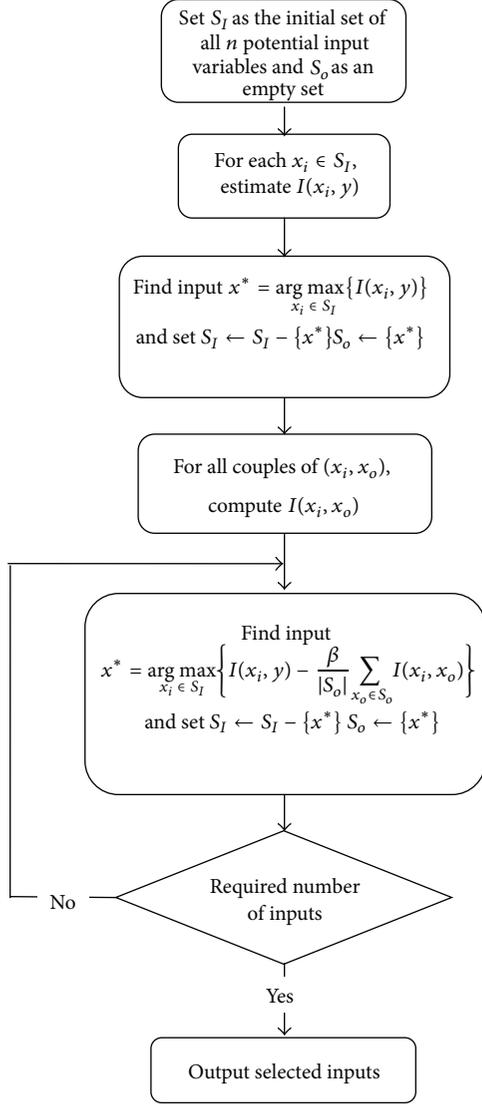


FIGURE 3: Operation of POLYMOT in the first four iterations in a two-dimensional input.

The training and test data sets and exogenous variables, introduced in Table 1, are identical to the data used by Amjady et al. [5] in order to make a fair comparison. For forecasting the wind power generation in each day of each test month,

the data from previous 50 days were used to train the LNF model using the POLYMOT algorithm.

Furthermore, for selection of the proper input variables, the wind power data of the last training day (i.e., the day prior



$S_I = \{x_1, \dots, x_n\}$  - set of all potential inputs  
 $S_o$ : set of selected inputs  
 $y$ : output variable  
 $\beta$ : input redundancy penalization factor

FIGURE 4: Procedure of MI-based input variable selection.

to the forecast day) was picked up as the validation data set. For this purpose, first the 50 lagged values of wind power, wind speed, and wind direction were fed into the MI-based input selection algorithm to sort the input variables in terms of their relevance to the output. Then, the most relevant input variable was fed to the LNF network, the model was trained using POLYMOT algorithm, and finally the validation error was calculated. Next, the second most relevant input variable was also added to the LNF model's input vector, the model was trained again, and the new validation error was calculated. This procedure was continued until the validation error attained its local minimum. The aforementioned input

TABLE 2: Selected input variables for forecasting  $WP_t$  for the first test day of April 2010.

Rank	Selected variables
1	$WP_{t-1}$
2	$WS_{t-1}$
3	$WS_{t-2}$
4	$WS_{t-3}$
5	$WP_{t-2}$
6	$WP_{t-6}$
7	$WD_{t-1}$
8	$WS_{t-12}$
9	$WS_{t-24}$
10	$WP_{t-24}$
11	$WD_{t-2}$
12	$WD_{t-12}$

Note:  $\{WP_t\}$ : wind power at time  $t$ ,  $\{WS_t\}$ : wind speed at time  $t$ , and  $\{WD_t\}$ : wind direction at time  $t$ .

selection procedure was repeated for each day in every test month.

In order to assess the performance of the proposed forecast methods, as well as perform reasonable comparisons, the root mean square error (RMSE) and mean absolute percentage error (MAPE) measures were used to evaluate accuracy of predictions.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (P_t - \hat{P}_t)^2}, \quad (12)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|P_t - \hat{P}_t|}{\bar{P}} \times 100, \quad (13)$$

$$\bar{P} = \frac{1}{N} \sum_{t=1}^N P_t, \quad (14)$$

where  $P_t$  and  $\hat{P}_t$  are the actual and forecasted wind power at hour  $t$ , respectively, and  $N$  is the number of predictions. In (13), the average value of wind power is selected in denominator (instead of actual value used in common definition of MAPE) to avoid the adverse effect of division by zero.

5.2. *Wind Power Forecasting for Sotavento Wind Farm.* The results of wind power forecasting for Sotavento wind farm are presented here. As the first step in construction of the forecast model, the proper input variables must be determined using the MI-based input selection algorithm outlined in Section 4. For a better illustration, the selected input variable as well as validation errors for the first day of April test month are presented here. Table 2 contains the selected input features for this test day, ranked based on their MI with the output. According to this table, 12 overall inputs have been selected for this test day. The value of validation error versus number of input variables is also shown in Figure 5. The lowest validation error not only shows the optimal number of input variables but also corresponds to the optimal value for the

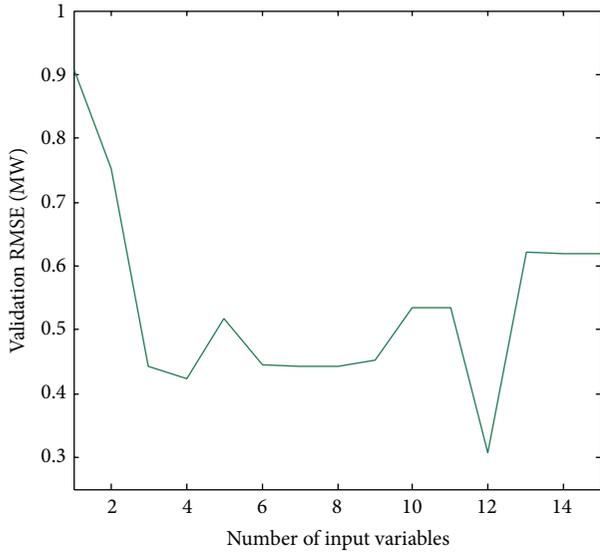


FIGURE 5: Validation error versus number of input variables (first day of April 2010).

parameters of the LNF model (i.e., local models and validity function parameters). In fact, structure of the best forecast model is determined using the validation data error. It should be noted here that for performing 24 step-ahead predictions, an iterative approach is adopted. That is, if in a test day the wind power of one hour ago ( $P_{t-1}$ ) is selected as model's input, the actual values of this input will be unknown when forecasting the wind power at hour 2 of the test day. In this case, the predicted value of wind power at hour 1 will be used as the input  $P_{t-1}$  for forecasting wind power at hour 2. This continues for the whole 24 hours of the test day.

In order to compare the performances of the POLYMOT and LOLIMOT learning algorithms, the training RMSE versus the number of neurons for both learning algorithms for the first test day of April 2010 are shown in Figure 6. It is clear that the POLYMOT algorithm finished the training procedure with 5 neurons and training RMSE of 0.18. However, the LOLIMOT algorithm constructed an LLNF network with 10 neurons and training RMSE of 0.28. This comparison shows the interesting features of the POLYMOT learning algorithm with respect to the LOLIMOT. It must be noted that the LOLIMOT and POLYMOT algorithms had the same input variables, stated in Table 2, in this comparison.

The actual and forecasted wind power for test months of April, May, June, and July are depicted in Figures 7, 8, 9, and 10, respectively. Clearly, the LNF approach has successfully followed the nonlinear variations in hourly wind power generation in all four test months. The peaks and valleys of the wind power generation have been accurately forecasted, indicating that the proposed LNF approach with POLYMOT learning algorithm and MI-based input selection has captured the nonlinear and complex nature of the wind power.

A comparative study with some of recently published methods is also of interest for better assessment of the LNF

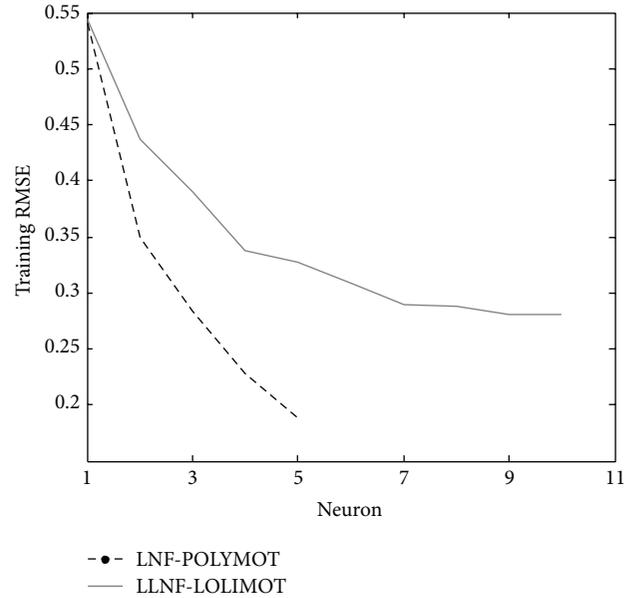


FIGURE 6: Comparison of training RMSE for LOLIMOT and POLYMOT learning algorithms for first day in April 2010.

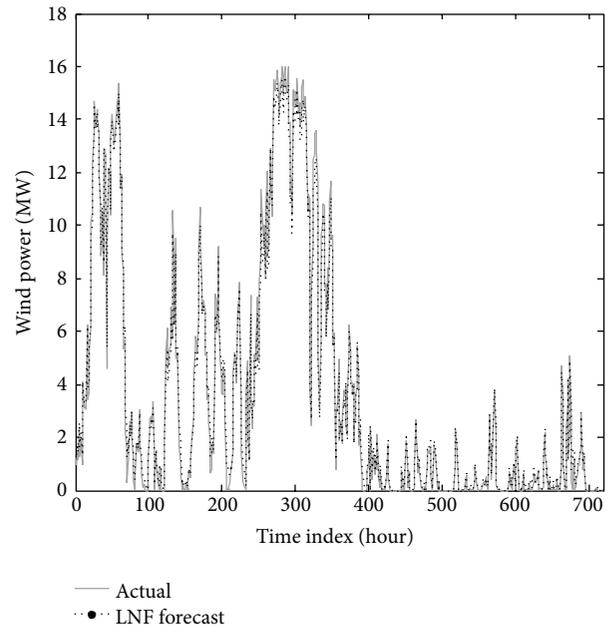


FIGURE 7: Actual and forecasted wind power for April 2010.

model accuracy. We considered persistence method, multivariate ARIMA, radial basis function (RBF) neural network, multilayer perceptron (MLP) neural network, and ridgelet neural network ridgelet (RNN), all developed by Amjady et al. in [5] for comparison since Amjady et al. used the same training and test data for their models. Moreover, the least-squares support vector machines (LSSVMs) are also used for comparison, as they are powerful time series prediction methods and have been employed for wind power forecasting [22]. The persistence method is a common benchmark

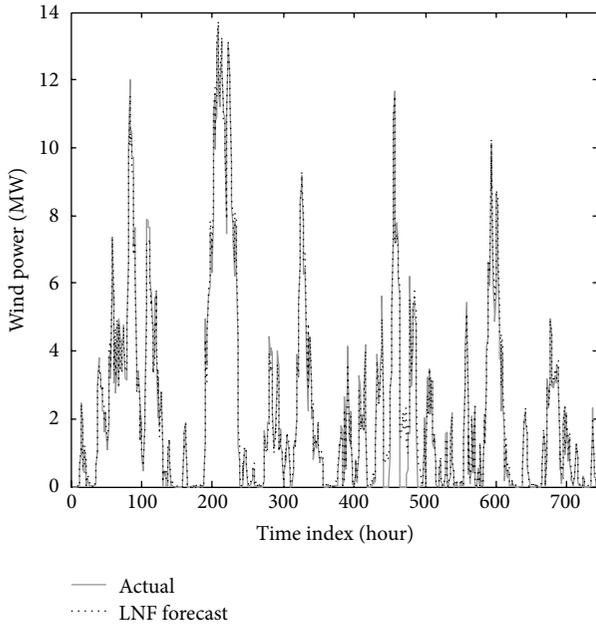


FIGURE 8: Actual and forecasted wind power for May 2010.

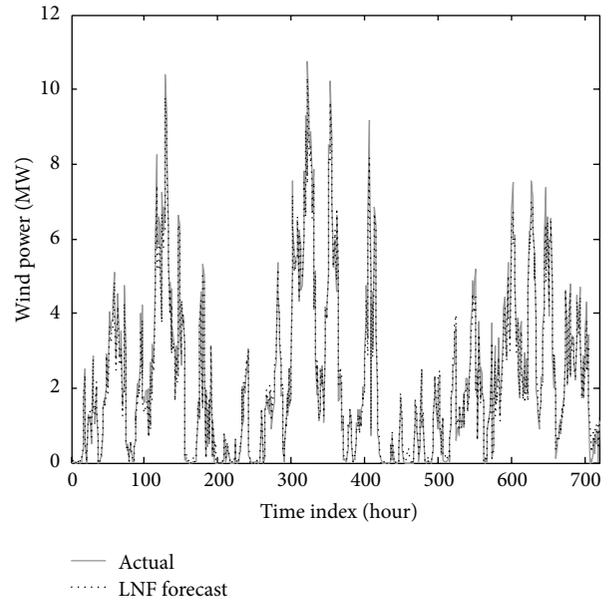


FIGURE 10: Actual and forecasted wind power for July 2010.

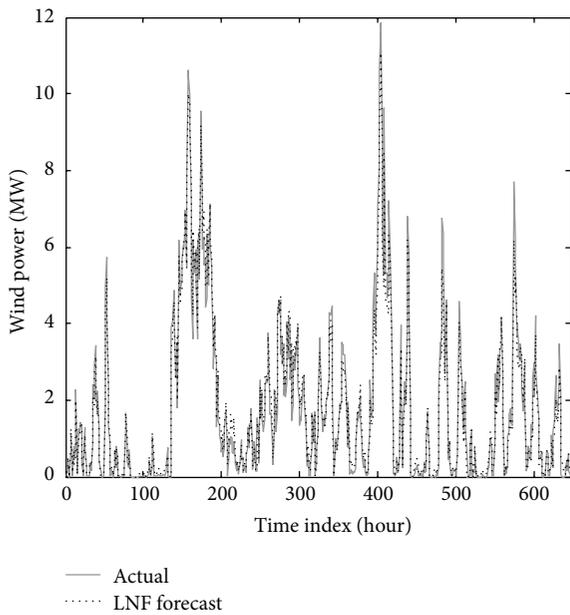


FIGURE 9: Actual and forecasted wind power for June 2010.

approach used for comparison in wind power forecasting [3, 4]. In the persistence method, the forecast for all future values in the forecast horizon is set to the last measured value [5]. The ARIMA which belongs to the category of classical time series techniques is a linear approach and does not perform well for complex and nonlinear processes. The RBF neural network can be considered as a special case of LNF models [18]. The MLP neural networks are good at capturing global data trends. However, increase of hidden layers in MLP may adversely affect generality of the model. The RNN

approach proposed in [5] is a three-layer neural network model with fixed weighting parameters between the hidden layer and output node and ridgelets as activation functions. The LSSVMs, which have been employed for wind power forecasting [22], are used for comparison as well. We also developed the local linear neurofuzzy (LLNF) model (note that if the order of polynomials in the LNF model is fixed at 1, then the LLNF model is resulted as a special case), trained by LOLIMOT algorithm, in order to evaluate improvement in LNF model's accuracy achieved through POLYMOT learning algorithm.

The comparison for all test months, in terms of RMSE and MAPE, are summarised in Tables 3 and 4, respectively. Interestingly, the proposed LNF model with POLYMOT learning algorithm has outperformed all other approaches for all test months. As stated earlier, the ARIMA model is linear and therefore it is expected that it cannot successfully model a nonlinear process. Both RBF neural network and LLNF model are special cases of the LNF model. Hence, it is justified to conclude that their performance never surpasses LNF model's performance. As presented in Table 3, the proposed LNF model has also shown higher accuracy with respect to the RNN approach. Although the ridgelet functions are appropriate basis set for constructing multivariate functions with certain kinds of spatial inhomogeneities and RNN can deal with a wide range of functions especially those with hyperplane singularities [5], the fixed connecting weights in RNN limit the model's flexibility and downgrade its performance, compared to the proposed LNF model, which has much more flexible polynomial functions as its local models. The obtained results show that the proposed LNF model also surpasses the LSSVM performance.

In order to present a clearer picture of the proposed LNF model's performance, the improvements in RMSE with respect to the compared methods and for all test months is

TABLE 3: Performance comparison based on RMSE.

Test month	Persistence method	Multivariate ARIMA	RBF	MLP	RNN	LSSVMs	LLNF	LNF
April 2010	1.24	0.843	0.549	0.514	0.463	0.49	0.485	0.328
May 2010	0.848	0.742	0.516	0.618	0.435	0.45	0.414	0.317
June 2010	0.784	0.702	0.593	0.521	0.437	0.39	0.428	0.299
July 2010	0.826	0.691	0.501	0.467	0.376	0.37	0.455	0.230
Average	0.895	0.744	0.551	0.530	0.428	0.442	0.445	0.293

TABLE 4: Performance comparison based on MAPE (%).

Test month	Persistence method	Multivariate ARIMA	RBF	MLP	RNN	LSSVMs	LLNF	LNF
April 2010	35.91	28.74	25.08	22.44	7.75	7.80	7.93	6.15
May 2010	30.84	27.21	18.11	19.82	11.43	11.55	10.12	8.29
June 2010	34.33	28.91	26.45	25.24	16.06	13.46	14.57	11.10
July 2010	36.84	29.54	27.75	26.14	9.33	9.25	10.03	7.38
Average	34.48	28.60	24.35	23.41	11.14	11.24	10.66	8.23

TABLE 5: RMSE improvement\* achieved by the proposed LLNF method with respect to the compared approaches.

Test month	Persistence method	Multivariate ARIMA	RBF	MLP	RNN	LSSVMs	LLNF
April 2010	73.5%	61.1%	40.3%	36.2%	29.2%	33.1%	32.4%
May 2010	62.6%	57.3%	38.6%	48.7%	27.1%	29.5%	23.4%
June 2010	61.9%	57.4%	49.6%	42.6%	31.6%	23.3%	30.1%
July 2010	72.2%	66.7%	54.1%	50.7%	38.8%	37.8%	49.5%
Average	67.3%	60.6%	46.8%	44.7%	31.5%	33.7%	34.2%

\*RMSE improvement = (RMSE of compared method – RMSE of LLNF)/(RMSE of compared method) × 1.

calculated and provided in Table 5. The RMSE improvement with respect to RNN method for April, May, June, and July 2010 is 29.2%, 27.1%, 31.6%, and 38.8%, respectively. This table also shows significant improvement against the LSSVMs. The improvement percentages demonstrate that the proposed LNF approach has greater capability of modelling nonlinear time series of wind power, since it incorporates more complicated polynomial functions instead of fixed connecting weights. RMSE improvement with respect to the LLNF model shows superiority of the POLYMOT learning algorithm over the LOLIMOT learning algorithm (used by LLNF model) in modelling complex processes.

## 6. Conclusion

This paper proposed an efficient approach for short-term wind power forecasting using local neurofuzzy models and POLYMOT learning algorithm. Starting from a single first-order polynomial model, the POLYMOT algorithm increases complexity of the model by employing high order polynomial or performing input space splitting, until the satisfactory performance is achieved. Use of high-order polynomial functions as local models in LNF model allows modelling highly nonlinear wind power time series in an efficient and accurate manner. Besides, with the help of MI-based feature selection algorithm, the most relevant input variables could be identified for wind power forecasting. The proposed approach was applied to wind power forecasting in Sotavento wind

farm, Spain, in four different months of 2010. The obtained forecasting results were satisfactory noticeable. Comparisons to some of recently published approaches demonstrated the superiority of the proposed LNF approach for wind power forecasting.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] International Energy Agency, World Energy Outlook, 2010, <http://www.iea.org/Textbase/about/copyright.asp>.
- [2] U.S. Energy Information Administration, <http://www.eia.gov/>.
- [3] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models," *IEEE Transactions on Energy Conversion*, vol. 24, no. 3, pp. 775–782, 2009.
- [4] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable Energy*, vol. 37, no. 1, pp. 1–8, 2012.
- [5] N. Amjady, F. Keynia, and H. Zareipour, "Short-term wind power forecasting using ridgelet neural network," *Electric Power Systems Research*, vol. 81, no. 12, pp. 2099–2107, 2011.
- [6] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Renewable Energy*, vol. 34, no. 5, pp. 1388–1393, 2009.

- [7] J. P. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes, "Hybrid intelligent approach for short-term wind power forecasting in Portugal," *IET Renewable Power Generation*, vol. 5, no. 3, pp. 251–257, 2011.
- [8] N. Amjady, F. Keynia, and H. Zareipour, "Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 3, pp. 265–276, 2011.
- [9] H. Zhou and J. Fang, "Wind power forecasting based on econometrics theory," in *Proceedings of the IEEE Electrical Power and Energy Conference*, pp. 1–6, Halifax, Canada, August 2010.
- [10] E. Cadenas and W. Rivera, "Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA-ANN model," *Renewable Energy*, vol. 35, no. 12, pp. 2732–2738, 2010.
- [11] R. W. Borg and L. J. Rothkrantz, "Short-term wind power prediction with radial basis networks," *Neural Network World*, vol. 16, no. 2, pp. 151–161, 2006.
- [12] G. Li, J. Shi, and J. Zhou, "Bayesian adaptive combination of short-term wind speed forecasts from neural network models," *Renewable Energy*, vol. 36, no. 1, pp. 352–359, 2011.
- [13] P. Louka, G. Galanis, N. Siebert et al., "Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, no. 12, pp. 2348–2362, 2008.
- [14] I. G. Damousis and P. Dokopoulos, "A fuzzy expert system for the forecasting of wind speed and power generation in wind farms," in *Proceedings of the 22nd IEEE International Conference on Power Industry Computer Applications (PICA '01)*, pp. 63–69, Sydney, Australia, May 2001.
- [15] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, and C. Draxl, "The state-of-the-art in short-term prediction of wind power: a literature overview," 2nd edition, Project report for the ANEMOS.plus and SafeWind projects, Roskilde, Denmark, 2011, <http://www.prediktor.dk/>.
- [16] C. Monterio, R. Bessa, V. Miranda, A. Botterud, J. Wang, and G. Conzelmann, *Wind Power Forecasting: State-of-the-Art 2009*, Argonne National Laboratory, 2009, <http://www.dis.anl.gov/pubs/65613.pdf>.
- [17] O. Nelles, *Nonlinear System Identification from Classical Approaches to Neural Networks and Fuzzy Models*, Springer, 2001.
- [18] O. Bänfer, M. Franke, and O. Nelles, "Adaptive local model networks with higher degree polynomials," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '10)*, pp. 168–171, Gyeonggi-do, Republic of Korea, October 2010.
- [19] H. Iranmanesh, M. Abdollahzade, and A. Miranian, "Mid-term energy demand forecasting by hybrid neuro-fuzzy models," *Energies*, vol. 5, no. 1, pp. 1–21, 2012.
- [20] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [21] "Sotavento wind farm data," <http://www.sotaventogalicia.com/>.
- [22] D. Liu and H. Li, "Short-term wind speed and output power forecasting based on WT and LSSVM," in *Proceedings of the International Conference on Information Engineering and Computer Science (ICIECS '09)*, pp. 1–4, December 2009.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

