Improved Short Term Wind Power Prediction Using A Combined Locally Weighted GMDH and KPCA

E. E. Elattar, *Member, IEEE*, I. Taha and Kamel A. Shoush

Abstract— Wind power prediction is one of the most critical aspects in wind power integration and operation. This paper proposes a new approach for wind power prediction. The proposed method is derived by integrating the kernel principal component analysis (KPCA) method with the locally weighted group method of data handling (LWGMDH) which can be derived by combining the GMDH with the local regression method and weighted least squares (WLS) regression. In the proposed model, KPCA is used to extract features of the inputs and obtain kernel principal components for constructing the phase space of the multivariate time series of the inputs. Then LWGMDH is employed to solve the wind power prediction problem. The coefficient parameters are calculated using the WLS regression where each point in the neighborhood is weighted according to its distance from the current prediction point. In addition, to optimize the weighting function bandwidth, the weighted distance algorithm is presented. The proposed model is evaluated using real world dataset. The results show that the proposed method provides a much better prediction performance in comparison with other models employing the same data.

Index Terms— Wind power prediction, group method of data handling, local predictors, locally weighted group method of data handling, weighted distance, kernel principal component analysis, state space reconstruction.

1 Introduction

WIND power is the fastest growing power generation sector in the world nowadays. The output power of wind farms is hard to control due to the uncertain and variable nature of the wind resources. Hence, the integration of a large share of wind power in an electricity system leads to some important challenges to the stability of power grid and the reliability of electricity supply [1]. Wind power prediction is one of the most critical aspects in wind power integration and operation. It allows scheduled operation of wind turbines and conventional generators, thus achieves low spinning reserve and optimal operating cost [2].

Short term prediction is generally for a few days, and hours to a few minutes, respectively. It is required in the generation commitment and market operation. Short term wind power prediction is a very important field of research for the energy sector, where the system operators must handle an important amount of fluctuating power from the increasing installed wind power capacity. Its time scales are in the order of some days (for the forecast horizon) and from minutes to hours (for the time-step) [3].

Various methods have been identified for short term wind power prediction. They can be categorized into physical methods, statistical methods, methods based upon artificial intelligence (AI) and hybrid approaches [4].

The physical method needs a lot of physical considerations to give a good prediction precision. It is usually used for long term prediction [5]. While the statistical performs well in short term prediction [6].

The traditional statistical methods are time-series-based methods, such as the persistence method [7], auto regressive integrated moving average (ARIMA) method [8], [9], etc. These methods are based on a linear regression model and can not always represent the nonlinear characteristics of the inputs. The AI methods describe the relation between input and output data from time series of the past by a non-statistical approach such as artificial neural network (ANN) [10], [11], fuzzy logic [7] and neuro-fuzzy [12]. Moreover, other hybrid methods [13], [14] have also been applied to short-term wind power prediction with success.

Support vector regression (SVR) [15] has been applied to wind speed prediction with success [16]. SVR has been shown to be very resistant to the overfitting problem and gives a high generalization performance in prediction problems. SVR has been evaluated on several time series datasets [17].

The Group Method of Data Handling (GMDH) is a selforganizing method that was firstly developed by Ivakhnenko [18].The main idea of GMDH is to build an analytical function in a feedforward network based on a quadratic node transfer function whose coefficients are obtained using a regression technique [19]. GMDH has been applied to solve many prediction problems with success [20], [21].

All the above techniques are known as global time series predictors in which a predictor is trained using all data available but give a prediction using a current data window. The global predictors suffer from some drawbacks which are discussed in the previous work [22].

- *This work was supported byTaif University, KSA under grant 2742-434-1.*
- *The Authors are with the department of the Electrical Engineering, Faculty of Engineering, Taif University, KSA.*
- *E. E. Elattar on leave from the Department of Electrical Engineering, Faculty of Engineering, Menofia University, Shebin El-Kom, Egypt (e-mail: dr.elattar10@yahoo.com).*
- *I. Taha is on leave from the Department of Electrical Power and Machines, Faculty of Engineering, Tanta University, Tanta, Egypt.*
- *Kamel A. Shoush is on leave from the Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Cairo, Egypt.*

The local SVR method is proposed by us to overcome the drawbacks of the global predictors [22]. More details of the local predictor can be found in [22]. Phase space reconstruction is an important step in local prediction methods. The traditional time series reconstruction techniques usually use the coordinate delay (CD) method [23] to calculate the embedding dimension and the time delay constant of the time series [24].

The traditional time series reconstruction techniques have a serious problem. In which there may be correlation betweendifferent features in reconstructed phase space. Consequently, the quality of phase space reconstruction and modelling will be affected [25]. In recent years, to process nonlinear time series, the kernel principal component analysis (KPCA) which is one type of nonlinear principal component analysis (PCA) is used [26]. KPCA is an unsupervised technique that is based on performing principal component analysis in the feature space of a kernel. The main idea of KPCA is first to map the original inputs into a high-dimensional feature space via a kernel map, which makes data structure more linear, and then to calculate principal components in the high-dimensional feature space [25].

Moreover, our previous work on local SVR predictor is extended to locally weighted support vector regression (LWSVR) by modifying the risk function of the SVR algorithm with the use of locally weighted regression (LWR) while keeping the regularization term in its original form [27], [28]. LWSVR has been applied to solve short term load forecasting (STLF) problem [27], [28]. Although LWSVR method improves the accuracy of STLF, it suffers from some limitations. First, the most serious limitation of SVR algorithm is uncertain in choice of a kernel. The best choice of kernel for a given problem is still a research issue. The second limitation is the selection of SVR parameters due to the lacking of the structural methods for confirming the selection of parameters efficiently. Finally, the SVR algorithm is computationally slower than the artificial neural networks.

To avoid the limitations of the existing methods and in order to follow the latest developments to have a modern system, a new method is proposed in this paper using an alternative machine learning technique which is called GMDH.

The proposed method is derived by combining the GMDH with the local regression method and weighted least squares regression and employing the weighted distance algorithm which uses the Mahalanobis distance to optimize the weighting function's bandwidth. In the proposed model, the phase space is reconstructed based on KPCA method, so that the problem of the traditional time series reconstruction techniques can be avoided. The proposed method has been evaluated using real world dataset.

The paper is organized as follows: Section 2 describes the time series reconstruction based on KPCA method. Section 3 reviews the GMDH algorithm. The LWGMDH method is introduced in Section 4. Section 5 describes the weighted distance algorithm. Experimental results and comparisons with other methods are presented in Section 6. Finally, Section 7 concludes the work.

2 Time Series Reconstruction Based on KPCA

The PCA is a well-known method for feature extraction [29]. It involves the computations in the input (data) space so it is a linear method in nature. KPCA is an unsupervised technique that is based on performing principal component analysis in the feature space of a kernel. KPCA can be used to reconstruct the time series, on the basis of which some kernel principal components are chosen according to their correlative degree to the model output to form final phase space of thenonlinear time series.

In KPCA the computations are performed in a feature space that is nonlinearly related to the input space. This feature space is that defined by an inner product kernel in accordance with Mercer's theorem [30]. Due to the nonlinear relationship between the input space and feature space the KPCA is nonlinear. However, unlike other forms of nonlinear PCA, the implementation of KPCA relies on linear algebra by mapping the original inputs into a high-dimensional feature space via a kernel map, which makes data structure more linear.

The basic idea of KPCA is to map the data x into a high dimensional feature space $\Phi(x)$ via a nonlinear mapping, and perform the linear PCA in that feature space:

$$Q(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \qquad (1)$$

where x_i and x_j are variables in input space and $Q(x_i, x_j)$ is called kernel function.

Given a set of data $X = \{x_i\}_{i=1}^{N}$ where each $x_i \in \Re^n$, we have a corresponding set of feature vector $\{\Phi(x_i)\}_{\mathrm{i=1}}^{\mathrm{N}}$. Accordingly, the sample covariance matrix $\Phi(x_i)$ can be defines as follows:

$$\widetilde{C} = \sum_{i=1}^{N} \Phi(x_i)\Phi^T(x_i) \qquad (2)$$

As in PCA method, we have to ensure that the set of feature vectors $\{\Phi(x_i)\}_{\mathrm{i=1}}^{\mathrm{N}}$ have zero mean [31]:

$$\frac{1}{\mathrm{N}} \sum_{i=1}^{N} \Phi(x_i) = 0 \qquad (3)$$

Proceeding then on the assumption that the feature vectors have been centered, KPCA solves the eigenvalues (4):

$$\lambda_i v_i = \tilde{C} v_i, \quad \mathrm{i=1,2,\ldots.., N} \qquad (4)$$

where λ_i is one of the non-zero eigenvalues of $\tilde{C}$ and v_i is the corresponding eigenvectors. Because the eigenvectors v_i in the plane which is composed of $\Phi(x_1), \Phi(x_2), \ldots.. \Phi(x_N)$. Therefore [25]:

$$\lambda_i \Phi(x_i) \cdot v_i = \Phi(x_i) \cdot \tilde{C} v_i, \quad \mathrm{i=1,2,\ldots.., N} \qquad (5)$$

And the exist coefficient α meet:

$$v = \sum_{j=1}^{N} \alpha_j \Phi(x_j) \qquad (6)$$

Substituting (2) and (6) into (4) and defining an $N \times N$ matrix Q which is defined by (1), the following formula can be got [30]:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_j \Phi(x_i) Q(x_i, x_j) = N\lambda \sum_{j=1}^{N}\alpha_j \Phi(x_j) \qquad (7)$$

Eq. (7) can be written in the compact matrix form [30]:

$$N\lambda\alpha = Q\alpha \qquad (8)$$

Assuming the eigenvectors of $\Phi(x_i)$ is of unit length v_i. $v_i = 1$, each α_i must be normalized using the corresponding eigenvalue by: . $\tilde{\alpha}_i = \frac{\alpha_i}{\sqrt{N\lambda_i}}, \quad i = 1,2,\ldots\ldots, N$

Finally the principal component for x_i, based on $\tilde{\alpha}_i$, can be calculated as following:

$$p_t(i) = v_i^T \Phi(x_i) = \sum_{j=1}^{N}\alpha_j \Phi(x_i, x_j), \quad i = 1,2,\ldots, N \qquad (9)$$

From (9), one can notice that the maximal number of principal components that can be extracted by KPCA is N. The dimensional of p_t can be reduced in KPCA by considering the first several eigenvectors that is sorted in descending order of the eigenvalues.

In this paper, we can employ the commonly used Gaussian kernel defined as:

$$Q(x_i, x) = \exp(-\frac{\|x_i - x\|}{2\sigma^2}) \qquad (10)$$

3 GROUP METHOD OF DATA HANDLING (GMDH)

Suppose that the original dataset consists of M columns of the values of the system input variables that is $X = (x_1(t), x_2(t), \ldots\ldots, x_M(t)), (t = 1,2,\ldots, N$ and a column of the observed values of the output and N is the length of the dataset.

The connection between inputs and outputs variables can be represented by a finite Volterra-Kolmogorov-Gabor polynomial of the form:

$$y = a_0 + \sum_{i=1}^{N} a_i x_i + \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} x_i x_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} a_{ijk} x_i x_j x_k + \ldots \qquad (11)$$

Where N is the number of the data of the dataset, *A*(a_0, a_i, a_{ij}, a_{ijk}, …..) and *X*(x_i, x_j, a_k, …..) are vectors of the coefficients and input variables of the resulting multi-input single-output system, respectively.

In the GMDH algorithm, the Volterra-Kolmogorov-Gabor series is estimated by a cascade of second order polynomials using only pairs of variables [18] in the form of:

$$\tilde{y} = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \qquad (12)$$

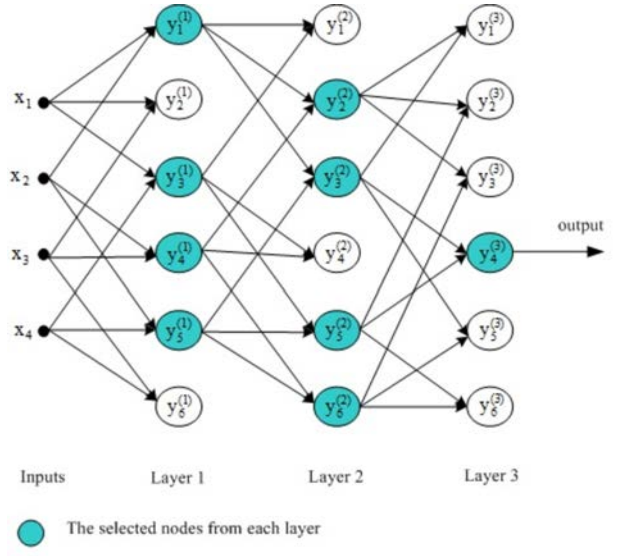

Fig. 1. GMHD network

The corresponding network as shown in Fig. 1 can be constructed from simple polynomial. As the learning procedure evolves, branches that do not contribute significantly to the specific output can be deleted; this allows only the dominant causal relationship to evolve.

The GMDH network training algorithm procedures can be summarized as follows:

- GMDH network begins with only input nodes and all combinations of different pairs of them are generated using a quadratic polynomial using (Eq. 12) and sent into the first layer of the network. The total number of polynomials (nodes) that can constructed is equal to *M(M-1)/2*.
- Use list squares regression to compute the optimal coefficients of each polynomial (node) $A(a_0, a_1, a_3, a_4, a_5)$ to make it best fit the training data as following:

$$e = \frac{1}{N}\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2 \qquad (13)$$

The least square solution of (Eq. 13) is given by:

$$A = (X^T X)^{-1} X^T Y \qquad (14)$$

Where, $Y = [y_1, y_2, \ldots\ldots\ldots, y_N]^T$,

$$X = \begin{bmatrix} 1 & x_{1P} & x_{1Q} & x_{1P}x_{1Q} & x_{1P}^2 & x_{1Q}^2 \\ 1 & x_{2P} & x_{1Q} & x_{2P}x_{2Q} & x_{2P}^2 & x_{2Q}^2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & x_{NP} & x_{NQ} & x_{NP}x_{NQ} & x_{NP}^2 & x_{QP}^2 \end{bmatrix}$$

and $P, Q \in \{1, 2, .., M\}$

- Compute the mean squared error for each node (Eq. 13).
- Sort the nodes in order of increasing error.
- Select the best nodes which give the smallest error from the candidate set to be used as input into the next layer with all combinations of different pairs of them being sent into second layer.
- This process is repeated until the current layer is found to not be as good as the previous one. Therefore, the previous layer best node is then used as the final solution.

More details about the GMDH and its different applications have been reported in [19, 31].

4 Locally Weighted Group Method of Data Handling (LWGMDH)

The LWGMDH method is derived by combining the GMDH with the local regression method and weighted least squares (WLS) regression. To predict the output values $\hat{y}$ for each query point (x_q) belongs to the testing set, the GMDH will be trained using the K nearest neighbors only (*1 < K << N*) of this x_q . The coefficient parameters is calculated using WLS regression where each point in the neighborhood is weighted according to its distance from the x_q . The points that are close to x_q have large weights and the points far from x_q have small weights

Overall, the framework of the design procedure of the LWGMDH comes as a sequence of the following steps.

- Step 1: Reconstruct the time series: load the multivariate time series dataset $X = (x_1(t), x_2(t_{),..} ....x_M(t))$, (t=1,2, …, N). Using the KPCA method to calculate the number of principal components of each dataset (we set the time delay constant of all datasets equal to 1). Then, reconstruct the multivariate time series using these values.
- Step 2: Form a training and validation data: The input dataset after reconstruction $\tilde{x}$ is divided into two parts, that is a training $\tilde{x}_{tr}$ dataset and validation $\tilde{x}_{va}$ dataset the size of the training dataset is N_{tr} while the size of the validation dataset is N_{va} .
- Step 3: For each query point x_q, choosing the *K* nearest neighbors of this query point using the Euclidian distance between x_q and each point in $X_{tr}(1 < K << N_{tr})$.
- Step 4: Create the first layer: using the K nearest neighbors only, all combinations of the inputs are generated based on (Eq. 12) and sent into the first layer of the network.
- Step 5: Estimate the coefficient parameters of each node: the vector of coefficient A is derived by minimizing the locally weighted mean squared error

$$e = \frac{1}{K} \sum_{i=1}^{K} w_i (y_i - \hat{y}_i)^2 \qquad (15)$$

where w is the weighting function. Meany weighting functions are proposed by the researchers [32]. Out of theses weighting functions, Gaussian kernel, tricube kernel and quadratic kernel are the most popular [32]. In this work, we employ the commonly used Gaussian kernel weighting function as following:

$$w_i = \sqrt{\exp(\frac{\left\| x_i - x_q \right\|^2}{h^2})} \qquad (16)$$

where x_q is the query point, x_i is a data point belongs the nearest neighbors points of x_q and *h* is the band width parameters which plays an important role in local modeling. An optimization method for the bandwidth is discussed in the next section in the paper. The weighted best square solution of (Eq. 15) is given by:

$$A = ((WX)^T (WX))^{-1} (WX)^T (Wy) \qquad (17)$$

where W is the diagonal matrix with diagonal elements $W_{ii} = w_i$ and zeros elsewhere [32], $y = [y_1, y_2, ..., y_K]^T$, $A = [a_0, a_1, a_2, a_3, a_4, a_5]$, X is defined in the last section but with number of rows equal to *K* (the number of the nearest) neighbors. This procedure is implemented repeatedly for all nodes of the layer.

- Step 6: Select the nodes with the best predictive capability to create the next layer: Each node in the current layer is evaluated using the training and validation datasets. Then the nodes which gives the best predictive performance for the output variable are chosen for input into the next layer with all combinations of the selected nodes based on (Eq. 12) being sent into next layer. In this paper, we use a predetermined number of these nodes. The coefficients parameters of each node in this layer can be estimated using the same procedures in step (5).
- Step 7: Check the stopping criterion: The modeling can be terminated when:

$$e_{l+1} \geq e_l \qquad (18)$$

where e_{l+1} is the minimal identification error of the current layer while e_l is a minimal identification error of the previous layer. So that the previous layer (l) best node is then used as the final solution of the current query point. If the stooping criterion is not satisfied, the model has to be expended. The steps 6 to 7 can be repeated until the stooping criterion is satisfied.

- Step 8: Then, the steps 3 to 7 can be repeated until the future values of different query points are all acquired.

Fig. 2 presents the computation procedure of the proposed method.

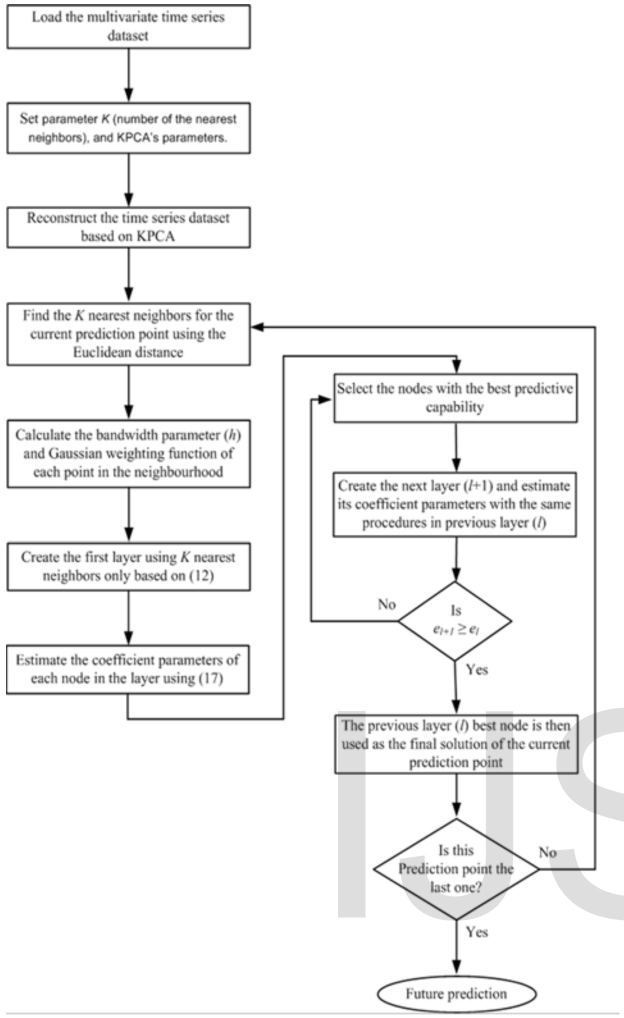
Load the multivariate time series dataset
Set parameter *K* (number of the nearest neighbors), and KPCA's parameters.
Reconstruct the time series dataset based on KPCA
Find the *K* nearest neighbors for the current prediction point using the Euclidean distance
Calculate the bandwidth parameter (*h*) and Gaussian weighting function of each point in the neighbourhood
Create the first layer using *K* nearest neighbors only based on (12)
Estimate the coefficient parameters of each node in the layer using (17)
Select the nodes with the best predictive capability
Create the next layer (*l*+1) and estimate its coefficient parameters with the same procedures in previous layer (*l*)
Is $e_{l+1} \geq e_l$
No
Yes
The previous layer (*l*) best node is then used as the final solution of the current prediction point
Is this Prediction point the last one?
No
Yes
Future prediction

Fig. 2. Flowchart of the proposed method

5 Weighted Distance Algorithm For Optimizing The Bandwidth

The weighting function bandwidth (*h*) is a very important parameter which plays an important role in local modeling. If *h* is infinite then the local modeling becomes global. On the other hand, if *h* is too small, then it is possible that we will not have an adequate number of data points in the neighborhood for a good prediction.

There are several ways to use this parameter like, constant bandwidth selection, nearest neighbor bandwidth selection where *h* is set to be a distance between the query point and the K^{th} nearest point, global bandwidth selection where *h* is calculated globally by an optimization process, etc [32].

The constant bandwidth selection method where training data with constant size and shape are used is the easiest and common way to adjust the radius of the weighting function. However, its performance is unsatisfactory for nonlinear system as density and distribution of data points are unlikely to be identical very place of the data set [33]. In this paper, we used the weighted distance algorithm which uses the Mahalanobis distance metric for optimizing the bandwidth (*h*) to improve the accuracy of our proposed method.

With the Mahalanobis distance metric, the problem of scale and correlation inherent in Euclidean distance are no longer an issue. In the Euclidean distance, the set of points which have equal distance from a given location is a sphere. The Mahalanobis distance metric stretches this sphere correct for the respective scales of the different variables.

The standard Mahalanobis distance metric can be defined as:

$$d(x)=\sqrt{(x-\mu)^T S^{-1}(x-\mu)} \qquad (19)$$

where x is the vector of data, μ is a mean and S^{-1} is inverse covariance matrix.

Defining the Mahalanobis distance metric between the query point x_q and data point x as $d_q=\sqrt{(x-x_q)^T S^{-1}(x-x_q)}$ where x belongs to the *K* nearest neighbors of the query point x_q and S^{-1} is computed after removing the main form each-column, the bandwidth is h_q the function of d_q :

$$h_q=\Theta(d_q) \qquad (20)$$

where $d_{\min} \le d_q \le d_{\max}$ and $d_{\min}$ is the distance between x_q and closest neighbor whileis $d_{\max}$is the distance between x_q and the farthest neighbor.

According to the LWR method, the query corresponding to $d_q=d_{\min}$ is most important that is $h_{\max}=\theta(d_{\min})=1$ while the query point corresponding to $d_q=d_{\max}$ is the least important, that is $h_{\min}=\theta(d_{\max})=\delta \cdot \delta$ is a real constant. This constant is a low sensitivity parameter. Therefore after few trails, we fix it to 0.01 which gives the best results.

The bandwidth $h_q$can be selected as a function of $d_q$as follows [33]:

$$h_q=\Theta(d_q)=a\ (\frac{1-bd_q}{d_q})+c \qquad (21)$$

where *a*, *b* and *c* are constants. By applying the boundary conditions, we can calculate these constants and get [33]:

$$h_q=(1-\delta)\left(\frac{d_{\min}(d_{\max}-d_q)}{d_q(d_{\max}-d_{\min})}\right)^2+\delta \qquad (22)$$

The Gaussian kernel weighting function which used in this paper can be written as following:

$$w=\sqrt{\exp(-\frac{d_q^2}{h_q^2})} \qquad (23)$$

6 Exprement Results

6.1 Data

To evaluate the performance of the proposed method, it has been tested for wind power prediction using the real data from wind farms in Alberta, Canada [34]. Alberta has the highest percentage of total installed wind generation capacity of any province in Canada. There are more than 40 wind projects proposed for future development in Alberta. Alberta includes many wind farms such as Ghost pine wind farm (owning 51 turbines and 81.6 MW total capacity), Taber wind farm (owning 37 turbines and 81.4 MW total capacity), Wintering Hills wind farm (owning 55 turbines and 88 MW total capacity), etc [35]. The total wind power installed capacity in 2011 is 800MW. This value will be raised to 893 MW by the most resent governmental goals for the wind sector in 2012 [35].

6.2 Parameters

To implement a good model, there are some important parameters to choose. There are two important parameters in the KPCA algorithm which used to reconstruct the phase space these parameters are the number of principal components (n_c) and w^2 in the Gaussian kernel function. The optimal values of these parameters which computed using the cross validation method are $w^2=1.9$ and $n_c=10$.

In the local prediction model, choosing the neighborhood size (K) is very important step. So, this parameters is calculated as describe in [27] where k_{max} and β are always fixed for all test cases at 45% of N and 80, respectively.

6.3 Forcasting Accuracy Evaluation

For all performed experiments, we quantified the prediction performance with root mean square error (RMSE) and normalized mean absolute error (NMAE) criterion. They can be defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{h=1}^{N}[\hat{p}_h - p_h]^2} \qquad (24)$$

$$NMAE = \frac{1}{N}\sum_{h=1}^{N}\left|\hat{p}_h - p_h\right| \times 100 \qquad (25)$$

where $\hat{p}_h$ and p_h are forecasted and actual electricity prices at hour h, respectively, p_{inst} is the installed wind power capacity and N is the number of forecasted hours.

6.4 Results and Discussion

The proposed LWGMDH method has been applied for the prediction of the whole wind power in Alberta, Canada. The performance of the proposed method in compared with 3 published approaches employing the same dataset. These approaches are resistance, seasonal ARIMA (SARIMA) and local radial basis function (LRBF). Historical wind power data are the only inputs for training the proposed method. For the sake of clear comparison, no exogenous variables are considered.

The proposed LWGMDH method predicts the value of the wind power subseries for one day ahead, taking into account the wind power data of the previous 3 months (the first 80% values of these data are used for training, while the last 20% values are used for validation). The length of the forecast horizon for the Alberta dataset is 24 hours. Four test weeks (Monday to Sunday) corresponding to four seasons of year 2011 are randomly selected for this numerical experiment. These test weeks are: the second week of February 2011 as a winter week, the third week of May 2011 as a spring week, the second week of August 2011 as a summer wee, and the first week November 2011 as a fall week.

The error (RMSE) and (NMAE) of each day during each testing week is calculated. Then the average error of each testing week (Monday to Sunday) is calculated by averaging the seven error values of its corresponding forecast days. Finally, the overall mean performance for the four testing weeks for each method can be calculated.

Table 1 shows a comparison between the proposed LWGMDH method and three other approaches (persistence, SARIMA and LRBF), reading the RMSE criterion. These results show that the proposed method outperforms other methods. Table 2 shows the RMSE improvements of the LWGMDH method over persistence, SARIME and LRBF. Table 3 shows a comparison between the proposed LWGMDH method and three other approaches (persistence, SARIMA and LRBF), regarding the NMAE criterion. These results show the superiority of the proposed method over other methods. Table 4 shows the NMAE improvements of the LWGMDH method over persistence, SARIMA and LRBF.

TABLE 1
Comparative RMSE Results

	Winter	Springs	Summer	Fall	Average
Persistence	13.71	16.19	14.42	22.99	16.83
SARIMA	6.70	6.59	8.09	13.88	8.82
LRBF	5.03	4.85	4.76	6.97	5.40
LWGMDH	4.01	3.90	3.72	5.32	4.24

TABLE 2
Improvement of The LWGMDH Over Other Approaches Regarding RMSE

	Average RMSE	Improvement
LWGMDH	4.24	--
Persistence	16.83	74.81%
SARIMA	8.82	51.93%
LRBF	5.40	21.48%

TABLE 3
Comparative NMAE Results

	Winter	Springs	Summer	Fall	Average
Persistence	6.59	7.66	7.51	11.07	8.21
SARIMA	3.21	3.09	3.84	6.53	4.17
LRBF	2.38	2.31	2.20	3.26	2.54
LWGMDH	1.94	1.85	1.71	2.48	1.99

TABLE 4
IMPROVEMENT OF THE LWGMDH OVER OTHER APPROACHES REGARDING NMAE

	Average RMSE	Improvement
LWGMDH	1.99	--
Persistence	8.21	75.76%
SARIMA	4.17	52.28%
LRBF	2.54	21.65%

Figs. 3- 6 show the predicted hourly wind power versus the actual wind power of one day (as an example) of each testing week using the proposed LWGMDH method. These results show that our prediction values are very close to the actual values.

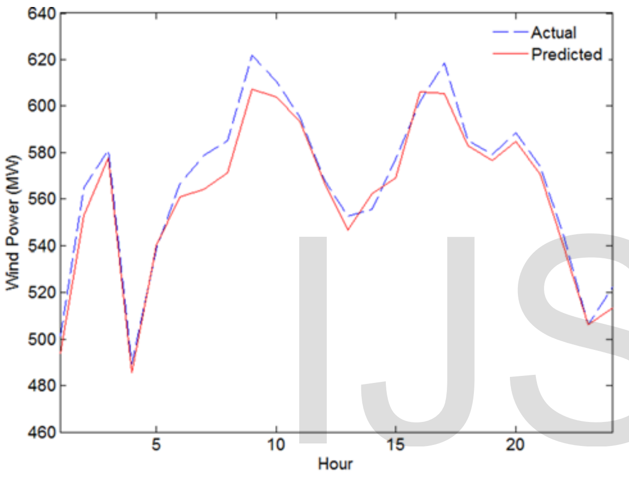

Fig. 3 Forecasted and actual hourly wind power for February 9, 2011

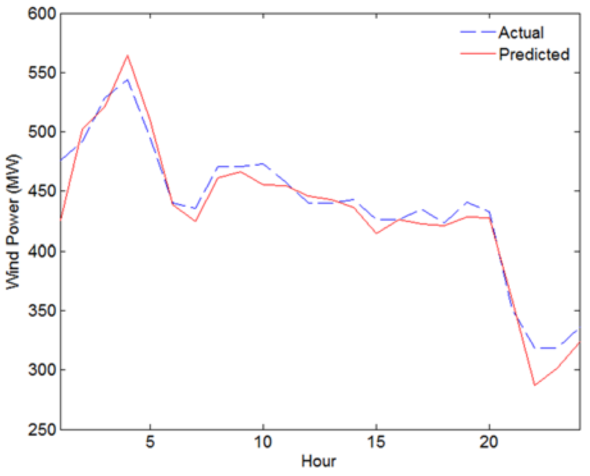

Fig. 4 Forecasted and actual hourly wind power for May 17, 2011

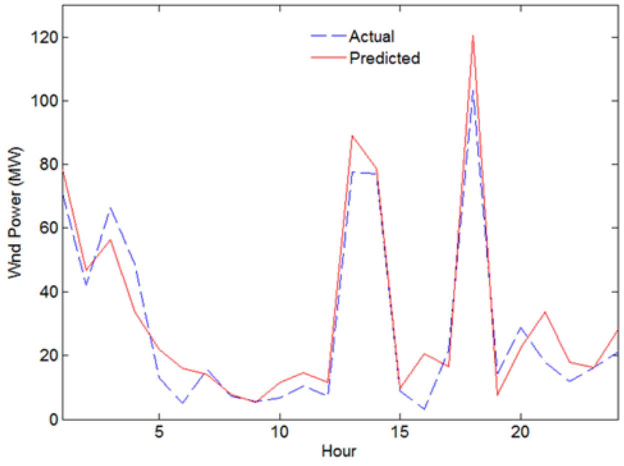

Fig. 5 Forecasted and actual hourly wind power for August 11, 2011

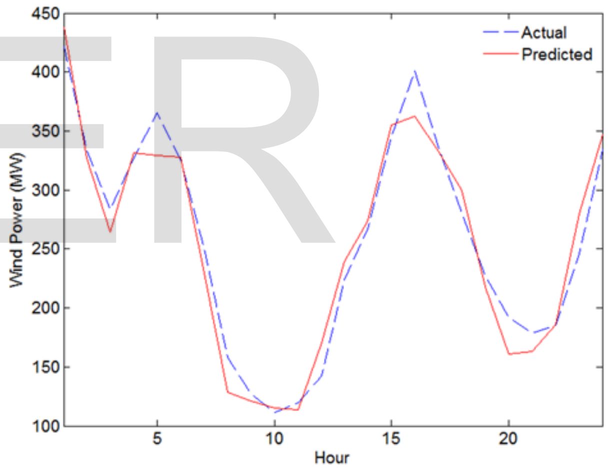

Fig. 6 Forecasted and actual hourly wind power for November 3, 2011

The above results indicate that the proposed LWGMDH method is less sensitivity to the wind power volatility than the other techniques used in the comparison.

To further study the superiority of LWGMDH method, it is also executed for all 52 weeks of year 2011 for the Alberta dataset and compared with three other approaches (Persistence, SARIMA and LRBF). The results show that the proposed LWGMDH method improves the RMSE and NMAE for the 52 weeks of year 2011 over the Persistence, SARIMA and LRBF methods.

Table 5 shows the RMSE and NMAE improvements of the LWGMDH method over Persistence, SARIMA and LRBF. In addition, Fig. 7 shows the comparison between LWGMDH method and Persistence, SARIMA and LRBF methods for each month of year 2011 regarding RMSE criterion. Same results can be got using the NMAE criterion.

These results show the robustness of the proposed LWGMDH method and its performance in a long run for a complete year.

TABLE 5
IMPROVEMENT OF THE LWGMDH OVER OTHER APPROACHES FOR ALL 52 WEEKS OF YEAR 2011

	RMSE Improvement	NMAE Improvement
LWGMDH	--	--
Persistence	73.88%	75.35%
SARIMA	51.19%	51.81%
LRBF	20.29%	21.01%

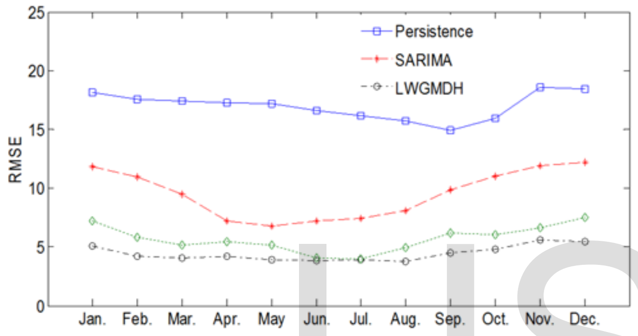

Fig. 7 RMSE Results for the Year Of 2011

7 CONCLUSION

In this paper, we have proposed a LWGMDH based KPCA method for wind power prediction. In the proposed method, the KPCA method is used to reconstruct the time series phase space and the neighboring points are presented by Euclidian distance for each query point. These neighboring points only can be used to train the GMDH where the coefficient parame ers are calculated using the weighted least square (WLS) regression. In addition, the weighting function's bandwidth which plays a very important role in local modelling is optimized by the weighted distance algorithm.

By using the KPCA the drawback of the traditional time series reconstruction techniques can be avoided by decreasing the correlation between different features in reconstructed phase space. Also, by combining GMDH with the local regression method the drawbacks of global methods can be overcome. In addition, by using the WLS, each point in the neighborhood is weighted according to its distance from the current query point. The points that are close to the current query point have larger weights than others. Moreover, by using the weighted distance algorithm, the disadvantage of using the weighting functions bandwidth as a fixed value can be overcome. This has led to improve the accuracy of the proposed model.

A real world dataset has been used to evaluate the performance of the proposed model which has been compared with Persistence, SARIMA and LRBF methods. The numerical results show the superiority of the proposed model over Persistence, SARIMA and LRBF methods based on different measuring errors.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Taif University for its support to carryout this work. It funded this project with a fund number 2742-434-1.

REFERENCES

[1] M. Mazadi, et al., “Impact of Wind Integration on Electricity Markets: a Chance-constrained Nash Cournot Model,” Int. Trans. on Electrical Energy Systems, vol. 23, no. 1, pp. 83–96, 2013.

[2] M. Khalid and A. Savkin, “A Method for Short-term Wind Power Prediction with Multiple Observation Points,” IEEE Trans. Power Systems, vol. 27, no. 2, pp. 579–586, 2012.

[3] H.M. I. Pousinho, V.M.F. Mendes, and J.P S. Catalo, “A hybrid PSO ANFIS Approach for Short-term Wind Power Prediction in Portugal,” Energy Conversion and Management, (In press.)

[4] M. Negnevitsky, P. Mandal, and A.K. Srivastava, “Machine Learning Applications for Load, Price and Wind Power Prediction in Power Systems,” Proc. of 15th Int. Conf. on Intelligent Syst. Appl. to Power Syst., Nov.8–12, pp. 1–6, 2009.

[5] L. Wang, L. Dong, Y. Hao, and X. Liao, “Wind Power Prediction Using Wavelet Transform and Chaotic Characteristics,” Proc. of World Non-Grid-Connected Wind Power and Energy Conf., (WNWEC 2009), Sept.24–26, pp. 1–5, 2009.

[6] L. Ma, et al., “A Review on the Forecasting of Wind Speed and Generated Power,” Renew. Sust. Energy Rev., vol. 13, no. 4, pp. 915–920, May 2009.

[7] G. Sideratos and N. Hatziargyriou, “Using Radial Basis Neural Networks to Estimate Wind Power Production,” Proc. of Power Eng. Soc. General Meeting, Tampa, FL, June24–28, pp. 1–7, 2007.

[8] W. Jiang, Z. Yan, D. Feng, and Z. Hu, “Wind Speed Forecasting Using Auto-Regressive Moving Average/Generalized Autoregressive Conditional Heteroscedasticity Model,” European Transactions on Electrical Power, vol. 22, no. 5, pp. 662–673, 2012.

[9] R.G. Kavasseri and K. Seetharaman, “Day-ahead Wind Speed Forecasting Using F-ARIMA Models,” Renew. Energy, vol. 34, no. 5, pp. 1388–1393, May 2009.

[10] 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 Trans. Sustainable Energy, vol. 2, no. 3, pp. 265–276, 2011.

[11] K. Bhaskar and S. Singh, “AWNN-assisted Wind Power Forecasting Using Feed-Forward Neural Network,” IEEE Trans. Sustainable Energy, vol. 3, no. 2, pp. 306–315, 2012.

[12] C.W. Potter and W. Negnevitsky, “Very Short-term Wind Forecasting for Tasmanian Power Generation,” IEEE Trans. Power Syst., vol. 21, no. 2, pp. 965–972, May 2006.

[13] N. Amjady, F. Keynia, and H. Zareipour, “A New Hybrid Iterative Method for Short-term Wind Speed Forecasting,” European Transactions on Electrical

Power, vol. 21, no. 1, pp. 581–595, 2011.

[14] S. Fan, et al., "Forecasting the Wind Generation Using a Two-stage Network Based on Meteorological Information," IEEE Trans. Energy Convers., vol. 24, no. 2, pp. 474–482, 2009.

[15] A. J. Smola and B. Scholkopf, "A Tutorial on Support Vector Regression," NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, 1998.

[16] D. Ying, J. Lu, and Q. Li, "Short-Term Wind Speed Forecasting of Wind Farm Based on Least Square-support Vector Machine," Power Syst. Technology, vol. 32, no. 15, pp. 62–66, Aug. 2008.

[17] M. Zhang, "Short-term Load Forecasting Based on Support Vector Machines Regression," Proc. of the Fourth Int. Conf. on Machine Learning and Cyber., 2005.

[18] A.G. Ivakhnenko, "Polynomial Theory of Complex Systems," IEEE Trans. of Syst., Man and Cyber., vol. SMC-1, pp. 364–378, 1971.

[19] S.J. Farlow, "Self-organizing Method in Modeling: GMDH Type Algorithm, Marcel Dekker Inc., 1984.

[20] D. Srinivasan, "Energy Demand Prediction Using GMDH Networks," Neurocomputing, vol. 72, pp. 625–629, 2008.

[21] R.E. Abdel-Aal, M.A. Elhadidy, and S.M. Shaahid, "Modeling and Forecasting the Mean Hourly Wind Speed Time Series Using GMDH-based Abductive Networks," Renewable Energy, vol. 34, no. 7, pp. 1686–1699, 2009.

[22] E.E. El-Attar, J.Y. Goulermas, and Q.H. Wu, "Forecasting Electric Daily Peak Load Based on Local Prediction," Proc. IEEE Power Engineering Society General Meeting (PESGM09), pp. 1–6, 2009.

[23] F. Takens, "Detecting Strange Attractors in Turbulence," Lect. Notes in Mathematics (Springer Berlin), vol. 898, pp. 366–381, 1981.

[24] D. Tao and X. Hongfei, "Chaotic Time Series Prediction Based on Radial Basis Function Network," Proc. Eighth ACIS Int. Conf. on Software Engin., Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 595–599, 2007.

[25] L. Caoa, et al., "A comparison of PCA, KPCA and ICA for Dimensionality Reduction in Support Vector Machine," Neurocomputing, vol. 55, pp. 321–336, 2003.

[26] F. Chen and C. Han, "Time Series Forecasting Based on Wavelet KPCA and Support Vector Machine," Proc. IEEE Int. Conf. on Automation and Logistics, pp. 1487–1491, 2007.

[27] E.E. Elattar, J.Y. Goulermas, and Q.H. Wu, "Electric Load Forecasting Based on Locally Weighted Support Vector Regression," IEEE Trans. Syst., Man and Cyber. C, Appl. and Rev., vol. 40, no. 4, pp. 438–447, 2010.

[28] E.E. Elattar, J.Y. Goulermas, and Q.H. Wu, "Integrating KPCA and Locally Weighted Support Vector Regression for Short-term Load Forecasting," Proc. the 15th IEEE Miditerranean Electrotechnical Conf. (MELECOn 2010), Valletta, Malta, Apr. 25–28, pp. 1528–1533, 2010.

[29] J. Xi and M. Han, "Reduction of the Multivariate Input Dimension Using Principal Component Analysis," Lect. Notes in Computer Science, vol. 4099, pp. 366–381, 2006.

[30] S. Haykin, "Neural networks: A Comprehensive Foundation", Printic-Hall, Inc., 1999.

[31] A.G. Ivakhnenko and G.A. Ivakhnenko, "The Review of Problems Solved by Algorithms of the Group Method of Data Handling (GMDH)," Pattern Recognition and Image Analysis, vol. 5, pp. 527–535, 1995.

[32] C.C. Atkeson, A.W. Moore, and S. Schaal, "Locally Weighted Learning," Artificial Intelligence Review (Special Issue on Lazy Learning), vol. 11, pp. 11–73, 1997.

[33] H. Wang, C. Cao, and H. Leung, "An Improved Locally Weighted Regression for a Converter Re-vanadium Prediction Modeling," Proc. the 6th World Congress on Intelligent Control and Automation, pp. 1515–1519, 2006.

[34] Alberta Electric System Operator (AESO), Wind Power tion http://www.aeso.ca/gridoperations/13902.html. (2013)

[35] Canadian Wind Power Association WEA). http://www.canwea.ca/farms/wind-farms e.php. (2013)