Genetic Programming Design of Locally Weighted GMDH for Wind Power Generation Prediction

E. E. Elattar, *Member, IEEE*, and I. Taha

Abstract— The intermittent nature of the wind creates significant uncertainty in the operation of power systems with increased wind power penetration. Considerable efforts have been made for the accurate prediction of the wind power considered one of the most rapidly growing sources of electricity generation all over the world. This paper proposes a new approach for wind power prediction. The proposed method is derived by integrating the genetic programming (GP) with locally weighted group method of data handling (LWGMDH). 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. The weighting function's bandwidth which plays a very important role in local modelling is optimized by the weighted distance algorithm Moreover, the GP algorithm is employed to optimize the structure of the GMDH network. Then optimized LWGMDH is employed to solve the wind power prediction problem. 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— Genetic programming, Wind power prediction, local predictors, optimized locally weighted group method of data handling, weighted distance, kernel principal component analysis, state space reconstruction.

1 INTRODUCTION

HIGH penetration of wind power in the electricity system provides a number of challenges to the grid operator, mainly due to the intermittency of wind. Since the power produced by a wind farm depends critically on the volatility of wind, unexpected variations of a wind farm output may increase operating costs for the electricity system by increased requirements of primary reserves, as well as place potential risks to the reliability of electricity supply [1,2]. However, such methods cannot guarantee accurate prediction of wind production variations; therefore, wind power forecasting tools become very important.

Various methods have been identified for wind power prediction. They can be categorized into physical methods, statistical methods, methods based upon artificial intelligence (AI) and hybrid approaches [3-7].

The Group Method of Data Handling (GMDH) is a self-organizing method that was firstly developed by Ivakhnenko [8].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 [9]. GMDH has been applied to solve many prediction problems with success [10], [11].

- *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.*

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. To overcome these drawbacks the local predictor based on support vector machine (SVR) method is proposed by us [12]. More details of the local predictors can be found in [13, 14]. To improve the accuracy of the proposed local predictors, the kernel principal component analysis (KPCA) [15, 16] method is used instead of the coordinate delay (CD) method [17] to phase space reconstruction which is an important step in local prediction methods [14].

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 by us using an alternative machine learning technique which is called locally weighted group method of data handling (LWGMDH) [18]. This 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 addition, the phase space is reconstructed based on KPCA method, so that the problem of the traditional time series reconstruction techniques can be avoided [18].

While providing a useful systematic design procedure, conventional GMDH used in LWGMDH [18] has some drawbacks [19]. First, it tends to generate quite complex polynomial for relatively simple systems (data). Second, it also tends to produce an exceedingly complex network when it comes to highly nonlinear systems due to its limited generic structure (quadratic two-variable polynomial) [20]. To overcome these drawbacks, a number of researchers have attempted to hybrid-

ise GMDH with some evolutionary optimisation techniques such as genetic algorithm (GA) [21] which showed that the proposed framework performs better than conventional GMDH method.

Genetic programming (GP) is viewed by many researchers as a specialisation of genetic algorithms. The population of genetic programming individuals is constructed as tree-structured expressions. The tree structure of individuals allows genetic programming to vary its size and shape, thus, achieving a high efficiency in searching of a solution space with respect to what genetic algorithms are able to do [22].

In order to solve the wind power prediction problem, a new design approach of LWGMDH based GP is proposed in this paper. In the proposed model, the phase space is reconstructed based on KPCA method. Then the weighted distance algorithm which uses the Mahalanobis distance is employed to optimize the weighting function's bandwidth. In addition, the GP algorithm used to design the whole architecture of the LWGMDH network. 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 LWGMDH algorithm. Section 4 describes the weighted distance algorithm. Genetic programming algorithmm is introduces in Section 5. Section 6 describes the GP design of GMDH. Then the GP design of LWGMDH is introdusec in Section 7. Experimental results and comparisons with other methods are presented in Section 8. Finally, Section 9 concludes the work.

2 Time Series Reconstruction Based on KPCA

In recent years, to process nonlinear time series, KPCA is used to overcome the CD method problem [16]. In KPCA the computations are performed in a feature space that is nonlinearly related to the input space. This feature space is defined by an inner product kernel in accordance with Mercer's theorem [23]. However, unlike other forms of nonlinear PCA, the implementation of KPCA relies on linear algebra by mapping the original inputs into a highdimensional feature space via a kernel map, which makes data structure more linear. In this paper, the commonly used Gaussian kernel is employed. The detail introduction of the basic KPCA can be viewed in [14, 23]

3 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 [18].

- 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 $\tilde{X}_{tr}$ $(1 < K << N_{tr})$.
- Step 4: Create the first layer: using the *K* nearest neighbors only, all combinations of the inputs are generated based on the following equation:

$$\tilde{y} = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \qquad (1)$$

where x belongs to the original dataset which consists of *M* columns of the values of the system input variables that is $X = (x_1(t), x_2(t), ..., x_M(t)), (t = 1,2,..., N)$ and a column of the observed values of the output and *N* is the length of the dataset. Then these combinations are sent into the first layer of the network.

- Step 5: Estimate the coefficient parameters of each node: the vector of coefficient $A(a_0, a_1, a_3, a_4, a_5)$ is derived by minimizing the locally weighted mean squared error

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

where *w* is the weighting function. Meany weighting functions are proposed by the researchers [24]. Out of theses weighting functions, Gaussian kernel, tricube kernel and quadratic kernel are the most popular [24]. In this work, we employ the commonly used Gaussian kernel weighting function as following:

$$w_i = \sqrt{\exp(\frac{\left\| x_i - x_q \right\|^2}{h^2})} \qquad (3)$$

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 laest square solution of (Eq. 2) is given by:

$$A = ((WX)^T (WX))^{-1} (WX)^T (Wy) \qquad (4)$$

where W is the diagonal matrix with diagonal elements $W_{ii} = w_i$ and zeros elsewhere [24], $y = [y_1, y_2, ..., y_K]^T$, $A = [a_0, a_1, a_2, a_3, a_4, a_5]$, 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. 1) 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:

$$e_{l+1} \geq e_l \quad (5)$$

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.

4 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 [24].

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 [25]. 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. More details about weighted distance algoritrhm can be found in [18].

5 Genetic Programming (GP)

Genetic programming (GP) is an evolutionary computation technique that automatically solves problems without having to tell the computer explicitly how to do it. At the most abstract level GP is a systematic, domain-independent method for getting computers to automatically solve problems starting from a high-level statement of what needs to be done.
Over the last decade, GP has attracted the interest of streams of researchers around the globe.

Technically, GP is a special evolutionary algorithm (EA) where the individuals in the population are computer programs. So, generation by generation GP iteratively transforms populations of programs into other populations of programs [26]. During the process, GP constructs new programs by applying genetic operations which are specialized to act on computer programs

Algorithmically, GP comprises the steps shown in Algorithm 1 [26]. The main genetic operations involved in GP (line 5 of Algorithm 1) are the following:

- Crossover: the creation of one or two offspring programs by recombining randomly chosen parts from two selected programs.
- Mutation: the creation of one new offspring program by randomly altering a randomly chosen part of one selected program.

Algorithm 1 Abstract GP algorithm.

1: Randomly create an initial population of programs from the available primitives.
2: **Repeat**
3: Execute each program and ascertain its fitness.
4: Select one or two program(s) from the population with a probability based on fitness to participate in genetic operations.
5: Create new individual program(s) by applying genetic operations with specified probabilities.
6: Until an acceptable solution is found or some other stopping condition is met (e.g., reaching a maximum number of generations).
7: R**eturn** the best-so-far individual.

In GP programs, the chromosomes are usually expressed as syntax trees rather than as lines of code. Figure 1 shows, for example, the tree representation of the program max (x*x, x+3*y).

Like in most other EAs, genetic operators in GP are applied to individuals that are probabilistically selected based on fitness. That is, better individuals are more likely to have more child programs than inferior individuals. The most commonly employed method for selecting individuals in GP is tourna-

ment selection [25].

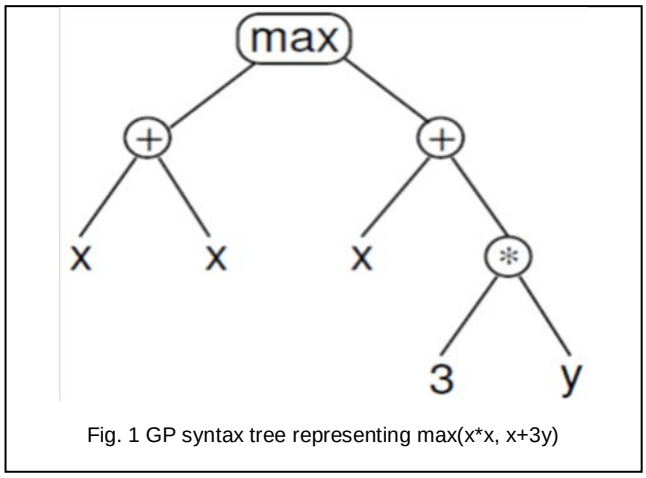

Fig. 1 GP syntax tree representing max(x*x, x+3y)

Given two indiviuals (parents), subtree crossover randomly selects a crossover point in each parent tree. Then, it creates the offspring by replacing the sub-tree rooted at the crossover point in a copy of the first parent with a copy of the sub-tree rooted at the crossover point in the second parent, as illustrated in Figure 2 [26].

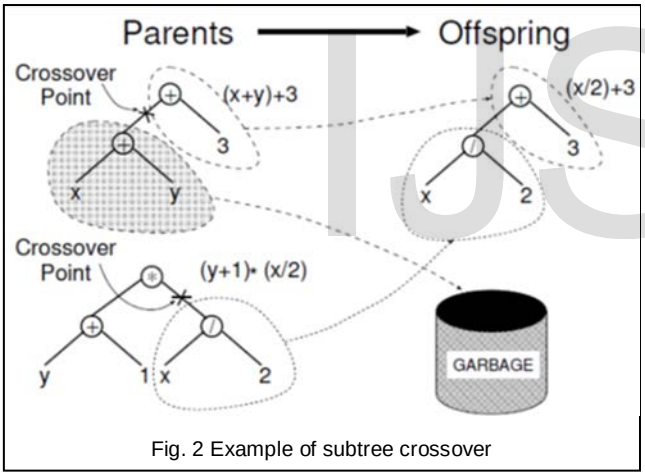

Fig. 2 Example of subtree crossover

The most commonly used form of mutation in GP (which we will call subtree mutation) randomly selects a mutation point in a tree and substitutes the sub-tree rooted there with a randomly generated sub-tree. This is illustrated in Figure 3 [26].

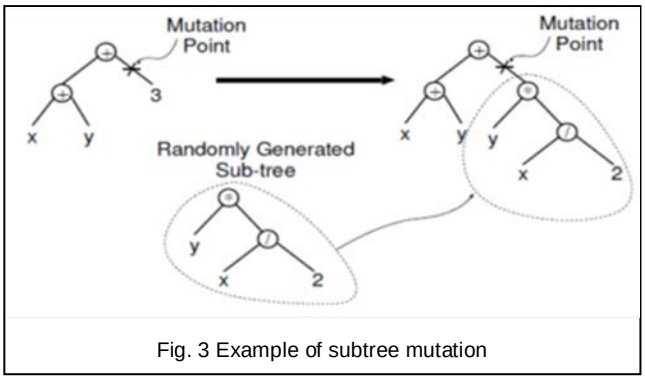

Fig. 3 Example of subtree mutation

6 Genetic Programming Design of GMDH

6.1 GP Algorithm

In a nutshell, GP algorithm is described below [27]:

- **Step1:** Initialize a population of tree expressions.
- **Step2:** Evaluate each expression in the population so as to derive the MDL-based fitness.
- **Step3:** Create new expressions (children) by mating current expressions. With a given probability, apply mutation and crossover to generate the child tree expressions.
- **Step4:** Replace the members of the population with the child trees.
- **Step5:** Execute the GMDH process, so as to compute the coefficients of the intermediate nodes of the child trees.
- **Step6:** If the termination criterion is satisfied, then halt; else go to Step2.

In Step 5, the coefficients of the child trees are recalculated using the GMDH process. However, this recalculation is performed only on intermediate nodes, upon whose descendant's crossover or mutation operators were applied. Therefore, the computational burden of Step5 is expected to be reduced as the generations proceed.

6.2 GMDH Process in GP

GP constructs a feedforward network, as it estimates the output function $\bar{f}$. The node transfer functions are simple (e.g. quadratic) polynomials of the two input variables, whose parameters are obtained using regression techniques.

As an example of a binary tree generated by GP, suppose that the upper left parent tree (P1) is written as a (Lisp) S-expression,

(NODE1
  (NODE2
    (NODE3 (x_1) (x_2))
    (x_3)
  (x_4)))

where x_1, x_2, x_3, x_4 are the input variables. Intermediate nodes represent simple polynomial relationships between two descendant (lower) nodes. This tree expresses a "complete form" $\bar{y}$ given by the GMDH process as follows:

1- Select two variables x_1 and x_2 and form an expression $G_{x1,x2}$ which approximates the output y (in terms of x_1 and x_2) with the least error using the multiple regression technique. Regard this function as a new variable z_1 (i.e. the new intermediate node NODE3),

$$z_1 = G_{x_1,x_2}(x_1, x_2) \tag{6}$$

2- Select two variables z_1 and x_3 and form an approximating expression $G_{z1,x3}$ in the same way. Regard this function as a new variable z_2 (i.e. the new intermediate node NODE2),

$$z_2 = G_{z_1,x_3}(z_1, x_3) \tag{7}$$

3- Select two variables z_2 and x_4 and form an approximating expression $G_{z2,x4}$. Regard this function as a "complete form" y, (i.e. the root node NODE1),

$$\bar{y} = G_{z_2,x_4}(z_2, x_4) \quad (8)$$

For the sake of simplicity, we assume quadratic expressions for the intermediate nodes. Thus each node records the information derived by the following equations:

$$NODE\,3: \quad z_1 = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 \quad (9)$$

$$NODE\,2: \quad z_2 = b_0 + b_1z_1 + b_2x_3 + b_3z_1x_3 + b_4z_1^2 + b_5x_3^2 \quad (10)$$

$$NODE\,1: \quad \bar{y}_1 = c_0 + c_1z_2 + c_2x_4 + c_3z_2x_4 + c_4z_2^2 + c_5x_4^2 \quad (11)$$

where z_1 and z_2 are intermediate variables, and $\bar{y}_1$ is an approximation of the output, i.e. the complete form. These equations are called "subexpressions".All coefficients (a_0, a_1, $\cdots$, c_5) are derived from multiple regression analysis using a given set of observations. For instance, the coefficients a_i in the equation (9) is calculated using the least square method as in equation (4). Suppose that *N* data triples (x_1,x_2,y) are supplied from observation, e.g.:

Note that all node coefficients are derived locally. For instance, consider bi's of NODE2. When applying the multiple-regression analysis to the equation (10), these bi's are calculated from the values of z_1 and x_3 (i.e. the two lower nodes), not from x_4 or y_1 (i.e. the upper node). Therefore, the GMDH process in GP can be regarded as a local-hill climbing search, in the sense that the coefficients of a node are dependent only on its two descendent (lower) nodes.

6.3 Crossover in GP

We now consider the recombination of binary trees in GP. Suppose two parent trees P1 and P2 are selected for recombination (Fig.4). Besides the above equations, internal nodes record polynomial relationships as listed below [27]:

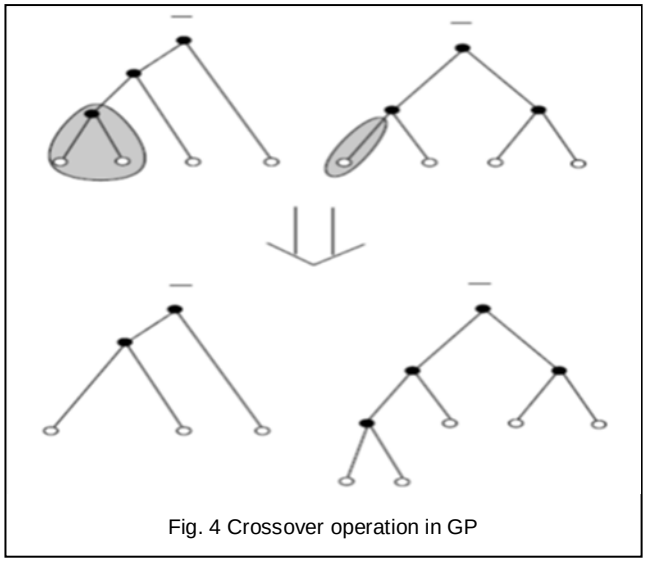

Fig. 4 Crossover operation in GP

$$NODE\,5: \quad z_3 = d_0 + d_1x_1 + d_2x_4 + d_3x_1x_4 + d_4x_1^2 + d_5x_4^2 \quad (12)$$

$$NODE\,6: \quad z_4 = e_0 + e_1x_3 + e_2x_1 + e_3x_3x_1 + e_4x_3^2 + e_5x_1^2 \quad (13)$$

$$NODE\,4: \quad \bar{y}_2 = f_0 + f_1z_3 + f_2z_4 + f_3z_3z_4 + f_4z_3^2 + f_5z_4^2 \quad (14)$$

Suppose z_1 in P1 and x_1 in P2 (shaded portions in Fig.4) are selected as crossover points in the respective parent trees. This gives rises to the two child trees C1 and C2 (lower part of Fig.4). The internal nodes represent the following relations:

$$NODE\,8: \quad \bar{y}_1' = b_0' + b_1'z_1' + b_2'x_4 + b_3'z_1'x_4 + b_4'z_1'^2 + b_5'x_4^2 \quad (15)$$

$$NODE\,7: \quad z_1' = a_0' + a_1'x_1 + a_2'x_3 + a_3'x_1x_3 + a_4'x_1^2 + a_5'x_3^2 \quad (16)$$

$$NODE\,12: \quad z_2' = c_0' + c_1'x_1 + c_2'x_2 + c_3'x_1x_2 + c_4'x_1^2 + c_5'x_2^2 \quad (17)$$

$$NODE\,10: \quad z_3' = d_0' + d_1'z_1' + d_2'x_4 + d_3'z_1'x_4 + d_4'z_1'^2 + d_5'x_4^2 \quad (18)$$

$$NODE\,11: \quad z_4' = e_0' + e_1'x_3 + e_2'x_1 + e_3'x_3x_1 + e_4'x_3^2 + e_5'x_1^2 \quad (19)$$

$$NODE\,9: \quad \bar{y}_2' = f_0' + f_1'z_3' + f_2'z_4' + f_3'z_3'z_4' + f_4'z_3'^2 + f_5'z_3'^2 \quad (20)$$

Since these expressions are derived from multiple regresion analysis, we have the following equations:

$$z_2' = z_1 \quad (21)$$

$$z_4' = z_4 \quad (22)$$

Thus, when applying crossover operations, we need only derive polynomial relations for $z_1', z_3', \bar{y}_1', \bar{y}_2'$. In other words, recalculation of the node coefficients for the replaced subtree (z_2') and non-replaced subtree (z_4') is not required, which reduces much of the computational burden in GP.

6.4 Mutation in GP

When applying mutation operations, we consider the following cases:

- A terminal node (i.e. an input variable) is mutated to another terminal node (i.e. another input variable).
- A terminal node (i.e. an input variable) is mutated to a nonterminal node (i.e. a subexpression).
- A nonterminal node (i.e. a subexpression) is mutated to a terminal node (i.e. an input variable).
- A nonterminal node (i.e. a subexpression) is mutated to another nonterminal node (i.e. another subexpression).

6.5 FitnessEvaluation in GP

GP uses a Minimum Description Length (MDL)-based fitness function for evaluating the tree structures. This fitness definition involves a tradeoff between certain structural details of the tree, and its fitting (or classification) errors [27].

MDL fitness = (Tree_Coding_Length) + (Exception_Coding_Length). (23)

The MDL fitness definition for our binary tree is defined as follows:

Tree_Coding_Length = 0.5 k log N (24)

Exception_Coding_Length = 0.5 N log S2N (25)

where N is the number of input-output data pairs, S_{2N} is the mean square error, i.e.

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N}\left|\bar{y}_i - y_i\right|^2 \quad (26)$$

and k is the number of parameters of the tree, e.g. the k-value for the tree P1 in Figure 4 is 6+6+6= 18 because each internal node has six parameters $(a_0, a_1, a_2, a_3, a_4, a_5)$ for NODE3 etc).

6.6 Overall Flow of GP

The GP algorithm is described below:

Input: tmax, I, Pop_size
Output: x, the best individual ever found.

1 t ← 0;
{I is a set of input variables. NODE 2 is a nonterminal node of 2-arity.}
2 P (t) ← initialize (Pop_size,I,{NODE_2});
3 F (t) ← evaluate (P (t),Pop_size);
4 x ← a_j(t) and Best_so_far ← MDL(a_j(t)), where MDL(a_j(t)) = min (F(t));
{the main loop of selection, recombination, mutation.}
5 **while** (ι (P (t), F (t), tmax) ≠ true) **do**
6 **for** i ←1 to do (Pop_size/2) **do**
{Select parent candidates using the MDL values.}
Parent1 ← select (P (t), F (t),Pop_size);
Parent2 ← select (P (t), F (t),Pop_size);
{Apply GP crossover operation.}
$a'_{2i-1}(t)$, $a'_{2i}(t)$ ← GP_recombine (Parent1, Parent2);
{Apply GP mutation operation, i.e. changing a node label and deleting/inserting a subtree.}
$d''_{2i}(t)$ ← GP_mutate ($a'_{2i}(t)$);
$d''_{2i-1}(t)$ ← GP_mutate ($a'_{2i-1}(t)$);
od
7 $P''(t)$ ← ($a''_1(t)$, · · ·, $a''_{pop\_size}(t)$);
8 F (t) ← evaluate($P''(t)$, Pop size);
9 tmp ← $a''_k(t)$, where MDL($a''_k(t)$ = min(F(t));
10 **if** (Best so far > MDL($a''_k(t)$))
then x ← tmp **and** Best so far ← MDL($a''_k(t)$));
11 P (t +1) ← $P''(t)$;
12 t ← t +1;
od
return (x);

{terminate if more than tmax generations are over.}

1 ι (P (t), F (t), tmax) :
2 **if** (t > tmax)
then return true;
else return false;

{initialize the population randomly.}

1 initialize (Pop_size, T, F):
2 **for** i ←1 **to** Pop_size **do**
generate a tree ai randomly,
where the terminal and nonterminal sets are T and F.
od
return (a_1, · · ·,a_{Pop_size});

{evaluate of a population of size Pop_size.}

1 evaluate (P (t), Pop_size):
2 **for** i ←1 **to** Pop_size **do**
{calculate eq.(26).}
GMDH Process (a_i);
S_N^2
(a_i) ← the mean square error of a_i;
{calculate eqs.(23),(24) and (25).}
MDL(a_i)←Tree Coding Length (a_i) + Exception Coding Length (a_i);
od
return (MDL(a_1), · · ·, MDL(a_{Pop_size}));

{execute the GMDH process.}

1 GMDH Process (a):
2 *nd* ← the root node of a;
3 **if** (*nd* is a terminal node)
then return;
{**if** the node coefficients of nd are already derived, **then return**.}
4 if (Coeff (*nd*) ≠ NULL)
then return;
5 *nl* ← left child (nd);
6 *nr* ← right child (*nd*);
7 GMDH Process (*nl*);
8 GMDH Process (*nr*);
9 Coeff (*nd*) ← Mult_Reg (*nl*,*nr*);
return;

7 Genetic Programming Design of LWGMDH

The conventional GMDH whiched used with GP in Section 6 has some drawbacks. These drawbacks have been discussed in Section 3. So, in order to solve the wind power prediction problem, a new design approach of LWGMDH based GP is introduced. In the proposed model, the phase space is reconstructed based on KPCA method. Then the weighted distance algorithm which uses the Mahalanobis distance is employed to optimize the weighting function's bandwidth. In addition, the GP algorithm used to design the whole architecture of the LWGMDH network.

Figure 5 presents the overall computation procedure

of the GP design of LWGMDH.

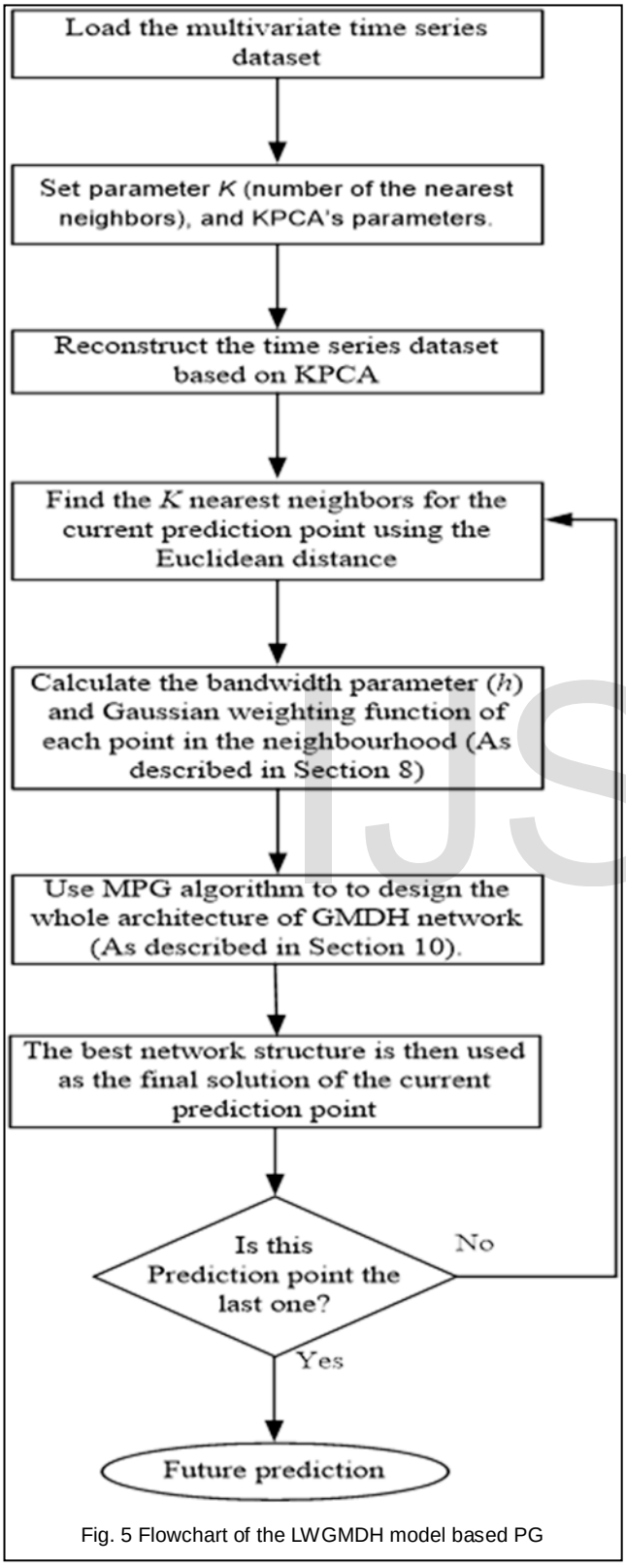

Fig. 5 Flowchart of the LWGMDH model based PG

8 EXPREMENT RESULTS

8.1 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 [28]. 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 [29]. 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 [29].

8.2 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 [13] where k_{max} and β are always fixed for all test cases at 45% of N and 80, respectively.

8.3 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:

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

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

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.

8.4 Results and Discussion

The proposed LWGMDH based GP (GP_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 4 published approaches employing the same dataset. These approaches are resistance, seasonal ARIMA (SARIMA), local radial basis function (LRBF) and LWGMDH. 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 GP_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 GP_LWGMDH method and four other approaches (persistence, SARIMA, LRBF and LWGMDH), reading the RMSE criterion. These results show that the proposed method outperforms other methods. Table 2 shows the RMSE improvements of the GP_LWGMDH method over persistence, SARIME, LRBF and LWGMDH. Table 3 shows a comparison between the proposed LWGMDH method and other approaches, regarding the NMAE criterion. These results show the superiority of the proposed method over other methods. Table 4 shows the NMAE improvements of the GP_LWGMDH method over other 4 methods.

Figs. 6- 9 show the predicted hourly wind power versus the actual wind power of one day (as an example) of each testing week using the proposed GP_LWGMDH method. These results show that our prediction values are very close to the actual values.

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
GP_LWGMD	3.33	3.50	2.99	4.92	3.69

TABLE 2
IMPROVEMENT OF THE GP_LWGMDH OVER OTHER APPROACHES REGARDING RMSE

	Average RMSE	Improvement
GP_LWGMDH	3.69	--
LWGMDH	4.24	12.97%
Persistence	16.83	78.08%
SARIMA	8.82	58.16%
LRBF	5.40	31.67%

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
GP_LWGMD	1.70	1.57	1.42	2.21	1.73

TABLE 4
IMPROVEMENT OF THE GP_LWGMDH OVER OTHER APPROACHES REGARDING NMAE

	Average RMSE	Improvement
GP_LWGMDH	1.73	--
LWGMDH	1.99	13.07%
Persistence	8.21	78.93%
SARIMA	4.17	58.51%
LRBF	2.54	31.89%

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

To further study the superiority of GP_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, LRBF and LWGMDH). The results show that the proposed GP_LWGMDH method improves the RMSE and NMAE for the 52 weeks of year 2011 over the Persistence, SARIMA, LRBF and LWGMDH methods.

Table 5 shows the RMSE and NMAE improvements of the GP_LWGMDH method over Persistence, SARIMA, LRBF and LWGMDH. In addition, Fig. 10 shows the comparison between GP_LWGMDH method and other 4 methods 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 GP_LWGMDH method and its performance in a long run for a complete year.

9 CONCLUSION

In this paper, we have proposed a LWGMDH based GP 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 parameters 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. In addition, the GP algorithm is employed to optimize the structure of the GMDH network.

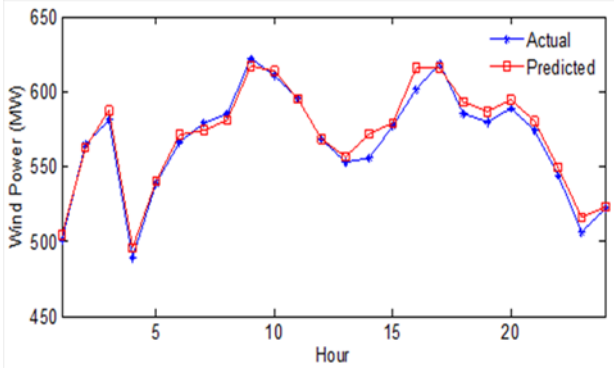

Fig. 6 Forecasted and actual hourly wind power for February 9, 2011

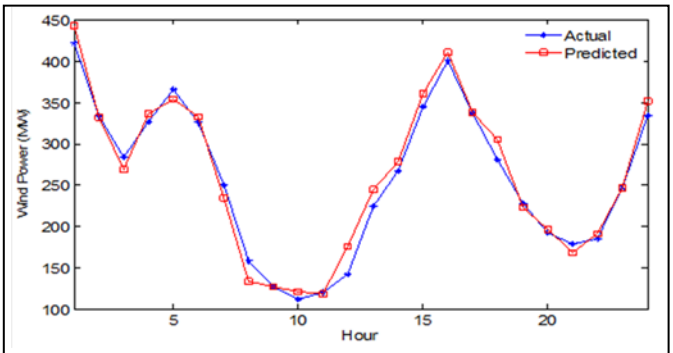

Fig. 9 Forecasted and actual hourly wind power for November 3, 2011

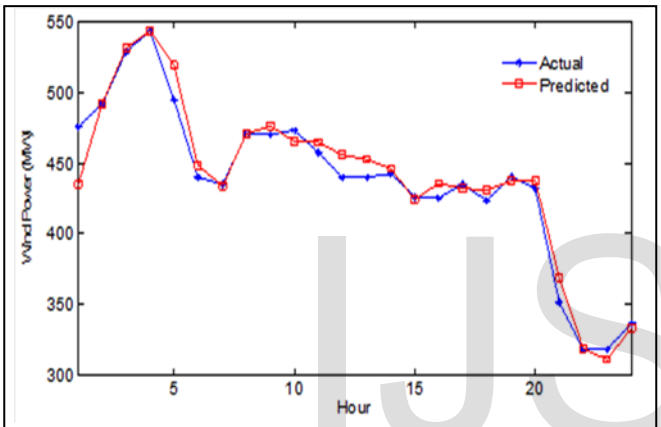

Fig. 7 Forecasted and actual hourly wind power for May 17, 2011

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

	RMSE Improvement	NMAE Improvement
GP_LWGMDH	--	--
LWGMDH	12.51%	12.72%
Persistence	77.98 %	78.30%
SARIMA	58.01 %	58.11%
LRBF	30.99%	31.26%

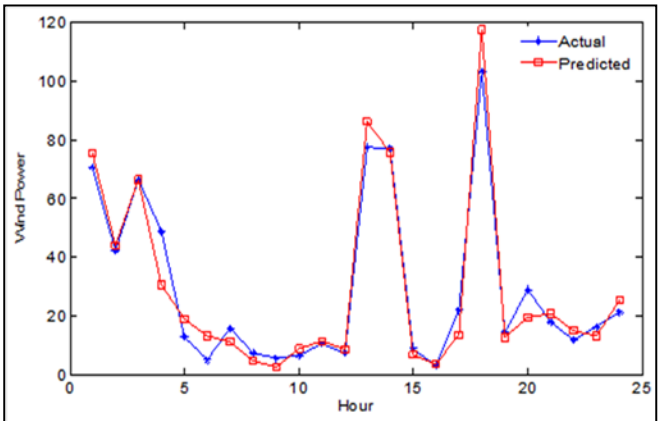

Fig. 8 Forecasted and actual hourly wind power for August 11, 2011

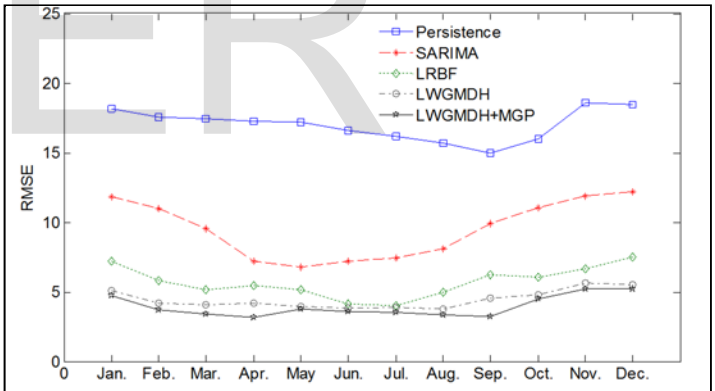

Fig. 10 RMSE Results for the Year Of 2011 for all methods

By using the KPCA the drawback of the traditional time series reconstruction techniques can be avoided. 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. 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.

Moreover, by using the GP with LWGMDH the drawbacks of conventional GMDH is overcome and the accuracy of the proposed algorithm has been improved.

A real world dataset has been used to evaluate the performance of the proposed model which has been compared with Persistence, SARIMA, LRBF and LWGMDH methods. The numerical results show the superiority of the proposed model over Persistence, SARIMA, LRBF and LWGMDH 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] B. Parsons, et al, "Grid Impacts of Wind Power: A Summary of Recent Studies in the United States," in Proc. EWEC, Madrid, Spain, 2003.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] E. E. Elattar, I. Taha and Kamel A. Shoush, "Improved Short Term Wind Power Prediction Using A Combined Locally Weighted GMDH and KPCA," Int. J. of Scientific & Engineering Research, vol. 4, No. 8, 2013.

[19] S. K. Oh. and W. Pedrycz. "The Design of Self-Organizing Polynomial Neural Networks:," Information Sciences, vol. 141, pp. 237-258, 2002.

[20] H. Park, B. Park, H. Kim, and S. Oh. "Self-organizing Polynomial Neural Networks Based on Genetically Optimized Multi-layer Perceptron Architecture", Int. Journal of Control, Automation and Systems, vol. 2, No. 4, pp. 423-434, 2004.

[21] N. Amanifard, et al., "Modelling and Pareto Optimization of Heat Transfer and flow Coefficients in Microchannels Using GMDH Type Neural Networks and Genetic Algorithms", Energy Conversion and Management, vol. 49, No.2, pp. 311-325, 2008.

[22] J. Koza, "Genetic programming", MIT press, 1998.

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

[24] 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.

[25] 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.

[26] R. Poli, W. Langdon, N. McPhee and J. Koza, "Genetic Programming: An Introductory Tutorial and a Survey of Techniques and Applications", Technical Report CES-475, ISSN: 1744-8050, 2007.

[27] I. Hitoshi, "Hybrid Genetic Programming and GMDH System: STROGANOFF", Studies in Computational Intelligence, Springer Verlag Berlin Heidelberg, vol. 211, pp. 27-98, 2011.

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

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