Short Term Load Forecasting Using Evolutionary Optimized Modified Locally Weighted GMDH

Ehab E. Elattar
Department of Electrical Engineering
Menofia University
Shebin El-Kom 32511, Egypt
dr.elattar10@yahoo.com

John Y. Goulermas, Q. H. Wu
Department of Electrical Engineering and Electronics
University of Liverpool
Liverpool L69 3GJ, UK
{j.y.goulermas, q.h.wu}@liverpool.ac.uk

Abstract - Accurate forecasting of electricity load is one of the most important issues in the electricity industry. It is essential part of an efficient power system planning and operation. Due to the limited generic structure of conventional group method of data handling (GMDH) network (quadratic two-variable polynomial), it tends to produce an exceedingly complex network when it comes to highly nonlinear systems. In order to overcome this, the modified locally weighted group method of data handling (M-LWGMDH) based genetic algorithm (GA) is proposed in this paper to solve the short term load forecasting problem. The LWGMDH is derived by combining the GMDH with local regression method and weighted least squares (WLS) regression. In the proposed method, the connectivity configuration is not limited to adjacent layers unlike the conventional GMDH. In addition, GA is used for optimal design of the M-LWGMDH network's topology where a new encoding scheme is presented. The hourly electricity load and temperature data in New York City are used to evaluate the proposed method. The results show that the proposed model exhibits superior performance to that of other methods.

Index Terms - Short term load forecasting, locally weighted group method of data handling, genetic algorithm, kernel principal component analysis, state space reconstruction.

I. INTRODUCTION

Short-term load forecasting (STLF) aims to predict electric loads for a period of minutes, hours, days, or weeks. STLF has always been a very important issue in economic and reliable power systems operation such as unit commitment, reducing spinning reserve, maintenance scheduling, etc. STLF is a difficult work because the accuracy of forecasting is influenced by many unpredicted factors such as economic, temperature, etc. whose relationships are commonly complex, implicit and nonlinear.

Many techniques have been proposed during the last few decades regarding STLF. The traditional models for STLF such as linear or multiple regression [1], autoregressive moving average exogenous variable (ARMAX) [2], etc cannot always represent the nonlinear characteristics of complex loads.

With the development of artificial intelligence techniques, artificial neural network [3], radial basis function network [4] and neuro-fuzzy model [5] are widely used in the STLF. These methods can deal with the nonlinear relation between the influencing factors and the load output, therefore the forecasting precision is raised. Support vector regression (SVR) which proposed based on statistical learning theory, has also been investigated as a promising approach to power load forecasting [6].

GMDH is a self-organizing method that was firstly developed by Ivakhnenko [7] as a multivariate analysis method for complex systems modelling and identification. 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 regression technique. GMDH has been applied to solve many prediction problems with success [8], [9]. In addition, many researchers have combined the GMDH with evolutionary methods such as genetic algorithm (GA) [10], particle swarm optimization [11], etc, to improve the accuracy of the conventional GMDH.

All the above techniques are known as global time series predictors in which a predictor is trained using all data available and used to predict the load at a specific time with a fixed data window. Our previous work has shown that the local prediction methods based on phase reconstruction can provide generally better results than those obtained with global methods based on phase reconstruction [12]. The local predictor involves more than one model to utilize the local information.

In this paper, we introduce a new design approach of GA based M-LWGMDH in order to solve the STLF problem. In load forecasting, it is common that some training points are more important than others. Therefore, the model should have higher accuracy for the training input data that are closer to the new input point for prediction. To achieve this goal and to overcome the drawbacks of global methods, we propose the M-LWGMDH method by combining the GMDH with local regression method and weighted least squares regression. Moreover, the connectivity configuration in the M-LWGMDH is not limited to adjacent layers unlike the conventional GMDH.

In the proposed method, the GA is used to design the whole architecture of M-LWGMDH network, i.e., the weighting function’s bandwidth, the number of nodes in each layer and their connectivity configuration. So, a new encoding scheme is introduced to present the genotype of general structure of the M-LWGMDH network. In addition, the phase space is reconstructed based on KPCA method so that the problem of the traditional time series reconstruction
techniques [13] can be avoided. The proposed method is evaluated using the hourly electricity load and temperature in New York City.

The paper is organized as follows: Section II describes the time series reconstruction based on KPCA method. Section III presents the LWGMDH algorithm. The GA based M-LWGMMDH method is introduced in Section IV. Experimental results and comparisons with other methods are presented in Section V. Finally, Section VI concludes the work.

II. SPACE RECONSTRUCTION OF MULTIVARIATE TIME SERIES BASED ON KPCA

Due to the complexity of the historical load data and the uncertainty of the influencing factors such as weather, economical, and random factors, the time series reconstruction technique can be applied to the power load forecasting. The traditional time series reconstruction techniques such as the coordinate delay (CD) method have a serious problem. This problem is that there may be correlation between different features in reconstructed phase space. This will influence the quality of phase space reconstruction and modelling effect. To overcome this problem, the kernel principal component analysis (KPCA) which is one type of nonlinear principal component analysis (PCA) is used recently to process the nonlinear time series.

Due to the nonlinear relationship between the input space and feature space the KPCA is nonlinear. 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 [14].

Suppose there is a set of data \( X = \{x_i\}_{i=1}^{N} \) where each \( x_i \in \mathbb{R}^n \) and the mean value \( E[X] = 0 \). By mapping \( x_i \) into \( \Phi \{x_i\} \), KPCA solves the eigenvalue (1) [13]:

\[
\lambda_i \upsilon_i = \tilde{C} \upsilon_i, \quad i = 1,2,...,N
\]  

(1)

where \( \tilde{C} = (1/N) \sum_{i=1}^{N} \Phi(x_i)\Phi(x_i)^T \) is the sample covariance matrix of \( \Phi \{x_i\} \), \( \lambda_i \) is one of the non-zero eigenvalues of \( \tilde{C} \) and \( \upsilon_i \) is the corresponding eigenvector.

Because the eigenvectors \( \upsilon_i \) in the plane which is composed of \( \Phi \{x_1\}, \Phi \{x_2\}, ..., \Phi \{x_N\} \). Therefore:

\[
\lambda_i \Phi(x_i) \cdot \upsilon_i = \Phi(x_i) \cdot \tilde{C} \upsilon_i, \quad i = 1,2,...,N
\]  

(2)

and there exist coefficients \( \alpha_{i}(i=1,2,...,N) \) meet:

\[
\upsilon_i = \sum_{j=1}^{N} \alpha_{i}(j) \Phi(x_j)
\]  

(3)

where \( \alpha_{i}(j) \) are the components of \( \alpha_{i} \). Combining (2) with (3) and defining an \( N \times N \) matrix \( Q \) (kernel function), the following formula can be got [13]:

\[
N\lambda_i \alpha_i = Q \alpha_i, \quad i = 1,2,...,N
\]  

(4)

Assuming the eigenvectors of \( \Phi \{x_i\} \) is of unit length \( \upsilon_i \cdot \upsilon_i = 1 \), each \( \alpha_i \) must be normalized using the corresponding eigenvalue \( \tilde{\alpha}_i = \alpha_i / \sqrt{\lambda_i} \), \( i=1,2,...,N \).

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

\[
p_i(i) = \upsilon_i^T\Phi(x_i) = \sum_{j=1}^{N} \tilde{\alpha}_i(j)Q(x_j,x_i), \quad i = 1,2,...,N
\]  

(5)

From (5), one can notice that the maximal number of principal components that can be extracted by KPCA is \( N \). The dimension of \( p_i \) can be reduced in KPCA by considering the first several eigenvectors that is sorted in descending order of the eigenvalues.

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

\[
Q(x_i,x) = \exp \left(\frac{-\|x_i - x\|^2}{2\sigma^2}\right)
\]  

(6)

III. LOCALLY WEIGHTED GROUP METHOD OF DATA HANDLING (LWGMDH)

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

The connection between inputs and output variables can be represented by an infinite Volterra-Kolmogorov-Gabor polynomial of the form:

\[
y = a_0 + \sum_{i=1}^{N} a_ix_i + \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}x_ix_j + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} a_{ijk}x_ix_jx_k + ...
\]  

(7)

where \( N \) is the number of the data of the dataset, \( A(a_0, a_i, a_{ij}, a_{ijk}, ...) \) and \( X(x_1, x_2, ..., x_M) \) are vectors of the coefficients and input variables of the resulting multi-input single-output system, respectively and \( y \) is the output of an individual node.

In the GMDH algorithm, the Volterra-Kolmogorov-Gabor series is estimated by a cascade of second order polynomials using only pairs of variables [7] in the form of:

\[
\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 + a_4 x_1^2 + a_5 x_2^2
\]  

(8)

where \( \hat{y} \) is the predicted output value.
When constructing the GMDH network, all combinations of the system input variables $X = x_1, x_2, ..., x_M$ are generated using (8) and sent into the first layer of the network. The coefficients of each node are derived using the least squares regression method. The best nodes between the outputs of this layer are selected for input into the second 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 neuron is then used as the final solution. More details about the GMDH and its different applications have been reported in [15].

The LWGMDH method is derived by combining the GMDH with the local regression method and WLS regression. To predict the output values by 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 are 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 $X = (x_1(t), x_2(t), ..., x_M(t)), (t = 1, 2, ..., N)$. Use 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 and output variables are chosen for input into the next layer with all combinations of the selected nodes based on (8) with all combinations of the selected nodes based on (8) being sent into next layer. The number of these nodes is predetermined by the user. The coefficients parameters of each node in this layer can be estimated using the same procedures in step 5.

- 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 $\tilde{X}_n$ ($1 < K << N$).

- Step 4: Create the first layer: Using the $K$ nearest neighbors only, all combinations of the inputs are generated based on (8) and sent into the first layer of the network.

- Step 5: Estimate the coefficients parameters of each node: The vector of coefficients $A$ is derived by minimizing the locally weighted mean squared error:

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

where $w$ is the weighting function, $y_i$ and $\hat{y}_i$ are the actual and output values, respectively. Many weighting functions are proposed by the researchers [16]. Out of these weighting functions, Gaussian kernel, tricube kernel and quadratic kernel are the most popular [17]. In this work, we used the Gaussian kernel weighting function as following:

$$ w_i = \exp \left( -\frac{||x_i - x_q||^2}{h^2} \right) \quad (10) $$

where $x_q$ is the query point, $x_i$ is a data point belongs to the nearest neighbors points of $x_q$ and $h$ is the bandwidth parameter which plays an important role in local modelling.

The weighted least square solution of (9) is given by:

$$ A = ((WX)^T (WX))^{-1} (WX)^T (Wy) \quad (11) $$

where $W$ is the diagonal matrix with diagonal elements

$$ W_{ii} = \sqrt{w_i} \text{ and zeros elsewhere} \quad [17], \ Y = [y_1, y_2, ..., y_K]^T, A(a_0, a_1, a_2, a_3, a_4), $$

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

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 nodes based on (8) being sent into next layer. The number of these nodes is predetermined by the user. 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 modelling can be terminated when:

$$ e_{l+1} \geq e_l \quad (12) $$

where $e_{l+1}$ is a 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 stopping criterion is not satisfied, the model has to be expanded. The steps 6 to 7 can be repeated until the stopping 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.

IV. TOPOLOGY DESIGN OF THE MODIFIED LWGMDH
Genetic Algorithms begin with a set of solutions (represented by population of chromosomes) created randomly. Following, the initial population is evaluated. Three main operators, reproduction, crossover and mutation, are used to evolve the initial population towards better solutions. The population is evaluated using a fitness function, and if the termination criteria are not met, the three main operators are applied again. One cycle of these operators and the evaluation procedure is known as a generation in GA terminology [18].

In the LWGMDH, the nodes in each layer are only connected to node in its adjacent layer. The M-LWGMDH removes this restriction. This means that the nodes in different layers including the input layer can be connected to others far away not only in the very adjacent layers. In the genetically design procedure of M-LWGMDH network, the most important consideration is the representation strategy, that is how to encode the different parameters into the chromosome. The encoding schemes in M-LWGMDH network must demonstrate the ability of representing different length and size of such networks.

In this paper, each chromosome which represents the structure of the whole network consists of two sub-chromosomes. We employ different coding methods for each sub-chromosome. The first sub-chromosome is a real coding for the bandwidth of the weighting function (h) while the second sub-chromosome is a string of alphabetical digits for the number of nodes in each layer and their connectivity configuration. The length of the second sub-chromosome is equal to \(2^{nl}\) where \(nl\) is the number of layers (hidden and output).

For example, for a given input vector \(x = (x_1, x_2, x_3, x_4)\), the second sub-chromosome can be represented as a string of 4 alphabetic symbols such as \(a, b, c,\) and \(d\). Therefore, a second sub-chromosome such as \(abacbd\) represents a unique structure topology of a M-LWGMDH network with 4 inputs for input layer, two hidden layers and single output \((2^{2+1} = 8)\) genes which is shown in Fig. 1.

As shown in Fig. 1, every part of the second sub-chromosome ababacbd with a length of \(2^1 = 2\) such as \(ab\) or \(bd\) or every part of the same sub-chromosome with a length of \(2^2 = 4\) such as abac – bd or every part of the same sub-chromosome with a length of \(2^3 = 8\) such as abacbd represents a particular neuron in the first hidden layer, second hidden layer and output layer, respectively.

Fig. 1 shows such connection of node \(bd\) directly to the output layer. It can be noticed that the output node (network’s output) includes \(bd\) twice. This means that a virtual node named \(bd\) has been reconstructed in the second hidden layer and used with \(abac\) to make the output neuron. It is noted that such repetition occurs when a node passes some adjacent hidden layers. The number this repetition depends on the number of passed hidden layers and can be calculated as \(2^{np}\) where \(np\) is the number of passed hidden layers.

The fitness function measures the performance of the model. It is quite important for evolving systems to find a good fitness measurement. The fitness \((F)\) of each entire string which represents a M-LWGMDH network is evaluated using mean square error defined as:

\[
F = \frac{1}{N_{va}} \sum_{i=1}^{N_{va}} (y_i - \hat{y}_i)^2
\]

where \(N_{va}\) is the number of points in the validation set, \(y_i\) and \(\hat{y}_i\) are the actual and output values, respectively.

Following this, the roulette wheel selection method is used for selecting two parents to produce two offspring through the crossover and mutation operators. For the first sub-chromosome which represents \(h\), the line arithmetical crossover [18] is used as follows:

\[
\begin{align*}
c_1 &= r \times h_1 + (1-r) \times h_2 \\
c_2 &= (1-r) \times h_1 + r \times h_2
\end{align*}
\]

where \(h_1\) and \(h_2\) are two parents, \(c_1\) and \(c_2\) are two offspring and \(r\) is a random generated uniform number \((r \in [0, 1])\).

The crossover operator for the second sub-chromosome is simply accomplished by exchanging the tail of each two sub-chromosomes from a randomly chosen point from the set \(\{2^1, 2^2, 2^3, ..., 2^{n-1}\}\). Fig. 2(a) shows an example of crossover operator for two chromosomes in M-LWGMDH networks. These networks after the crossover are shown in Fig. 2(b).

Similarly, the mutation operation can contribute effectively to the diversity of the population. In the first sub-chromosome, the Gaussian mutation [19] is used. While this operation is simply accomplished by changing one or more digits as genes in the second sub-chromosome to another possible digit. After that, each of the population generated goes through a series of evaluation, reproduction, crossover and mutation. In every generation, the chromosome that has the worst fitness value is replaced by the chromosome that has
Fig. 2. Crossover operation for a modified LWGMDH: (a) Chromosomes, and (b) Modified LWGMDH networks.

the best fitness value in the previous generation. These procedures are repeated until a termination condition is reached. In this paper, we use a predetermined maximum number of generations as a termination condition. After the termination condition is satisfied, the chromosome which gives the best performance in the last generation is selected as the output M-LWGMDH network.

V. MODIFIED LWGMDH FOR SHORT TERM LOAD FORECASTING

To evaluate the performance of the proposed method, the hourly electricity load in New York City and temperature data observed at Central Park have been considered in this paper [20].

Choosing $K$ is very important step in order to establish the local prediction model. There are some methods used in literatures to find this parameter such as cross validation [21] and bootstrap [22]. This parameter should be high for low density datasets while it should be low for high density ones. So, in this paper, we calculate $K$ by designing a systematic method as follows [23]:

$$K = \text{round} \left( \frac{\alpha}{N \times k_{\text{max}} \times D_{\text{max}}} \sum_{i=1}^{N} \sum_{k=1}^{k_{\text{max}}} D_k(x_i) \right)$$  \hspace{1cm} (15)

where $N$ is the number of training points, $k_{\text{max}}$ is the maximum number of nearest neighbors, $D_k(x_i)$ is the distance between each training point $x$ and its nearest neighbors while $D_{\text{max}}$ is the maximum distance, 

$$\frac{1}{N \times k_{\text{max}} \times D_{\text{max}}} \sum_{i=1}^{N} \sum_{k=1}^{k_{\text{max}}} D_k(x_i)$$ is the average distance around the points which is inversely proportional to the local densities and $\alpha$ is a constant. The two constants $k_{\text{max}}$ and $\alpha$ are very low sensitivity parameters. $k_{\text{max}}$ can be chosen as a percentage of the number of training points ($N$) for efficiency while $\alpha$ can be chosen as a percentage. In this paper, $k_{\text{max}}$ and $\alpha$ are always fixed for all test cases at 50\% of $N$ and 90\%, respectively.

In addition, the parameters of the KPCA algorithm which are the number of principal components ($n_c$) and $\sigma^2$ in the Gaussian kernel function are computed using the cross validation method. The optimal values of these parameters are shown in Table I. Also, Table I summarizes the design parameters of GA.

We quantified the prediction performance with the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). They can be defined as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| A(i) - F(i) \right|$$  \hspace{1cm} (16)

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{A(i) - F(i)}{A(i)} \right) \times 100$$  \hspace{1cm} (17)

where $A$ and $F$ are the actual and the forecasted loads, respectively, $n$ is the testing dataset size, and $i$ denotes the test instance index.

To verify the predictive ability of the proposed method, its performance is compared with GMDH (as a global method), LWSVR which is proposed by us in [23], the prediction of New York Independent System Operator (NYISO) [20] and an adaptive two-stage hybrid network with self-organized map (SOM) and support vector machine (SVM) [24]. To make results comparable with other methods, we used the same experimental setup as used in [24]. That is two typical months have been used for forecasting. The first one corresponds to January 2004, which is a winter month while the second one corresponds to July 2004, which is a summer month. The hourly load and temperature data from 1st January 2003 to 31st December 2003 are used to forecast the first testing month. Whereas, the hourly load and temperature data from 1st July

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>DESIGN PARAMETERS OF THE PROPOSED METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>GA</td>
</tr>
<tr>
<td>Maximum generation</td>
<td>100</td>
</tr>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.8</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.07</td>
</tr>
<tr>
<td>Weighting function’s bandwidth</td>
<td>$0 &lt; h \leq 1$</td>
</tr>
<tr>
<td>$n_c$</td>
<td>15</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>1.30</td>
</tr>
</tbody>
</table>
2003 to 30th June 2004 are used to forecast the second testing month.

First, we calculate the MAE and MAPE of each day during the two test months. Then the average MAE and MAPE values of each method for the two test months are calculated. Also, we calculate the MAE and MAPE of the forecasting load published by NYISO [20] in the same period. These results are shown in Table II for MAE and Table III for MAPE.

These results show that the proposed method outperforms other methods. It improves the MAPE over NYISO [20], hybrid network [24], GMDH, and LWSVR by 61.19%, 39.01%, 44.78% and 20.71%, respectively for the winter month and 59.15%, 36.68%, 42.91% and 18.54%, respectively for summer month.

Another observation from these results is that the MAPE of July (summer month) is higher than January (winter month). This happens because in New York City July is unstable month in respect to load behavior. This is due to the increase in power consumption because of a gradual rise in temperature.

Figs. 3 and 4 show the actual load and forecasted load values using our proposed method of one week, as an example, of each testing month. These results show that our proposed method’s prediction values are very close to the actual values.

Moreover, for each method, the average MAPE value of the two test months for each day of the week (Monday to Sunday) is calculated. These results are shown in Fig. 5. Again, these results confirm the superiority of the proposed method over other methods.

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>OVERALL MAE OF EACH METHOD FOR EACH TESTING PERIOD (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction model</td>
<td>Winter month</td>
</tr>
<tr>
<td>NYISO [20]</td>
<td>163.05</td>
</tr>
<tr>
<td>Hybrid network [24]</td>
<td>106.97</td>
</tr>
<tr>
<td>GMDH</td>
<td>123.04</td>
</tr>
<tr>
<td>LWSVR</td>
<td>75.01</td>
</tr>
<tr>
<td>Proposed method</td>
<td>64.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>OVERALL MAPE OF EACH METHOD FOR EACH TESTING PERIOD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction model</td>
<td>Winter month</td>
</tr>
<tr>
<td>NYISO [20]</td>
<td>2.86</td>
</tr>
<tr>
<td>Hybrid network [24]</td>
<td>1.82</td>
</tr>
<tr>
<td>GMDH</td>
<td>2.01</td>
</tr>
<tr>
<td>LWSVR</td>
<td>1.40</td>
</tr>
<tr>
<td>Proposed method</td>
<td>1.11</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

In this paper, we have proposed a new approach which can be used to solve STLF problem. In the proposed method, the KPCA method has been used to reconstruct the phase space of time series so that the drawbacks of the traditional time series reconstruction techniques can be overcome. In addition, by combining the local regression method and WLS regression with GMDH, the disadvantages of the global methods can be avoided. Also each point in the neighborhood is weighted according to its distance from the current query
To verify the predictive ability of the proposed method, we used the historical load and temperature data in New York City. The proposed method has been compared with the LWSVR, conventional GMDH and published models employing the same dataset. The numerical results show the superiority of the proposed method over all other methods.

REFERENCES


