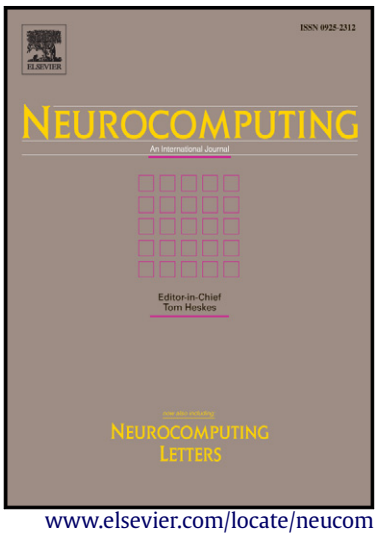


www.elsevier.com/locate/neucom

Hierarchical Extreme Learning Machine for Feedforward Neural Network

Hong-Gui Han, Li-Dan Wang, Jun-Fei Qiao

PII: S0925-2312(13)00733-9
DOI: http://dx.doi.org/10.1016/j.neucom.2013.01.057
Reference: NEUCOM13528

To appear in: *Neurocomputing*

Received date: 26 August 2012
Revised date: 15 December 2012
Accepted date: 3 January 2013

Cite this article as: Hong-Gui Han, Li-Dan Wang, Jun-Fei Qiao, Hierarchical Extreme Learning Machine for Feedforward Neural Network, *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2013.01.057

Hierarchical Extreme Learning Machine for Feedforward Neural Network

Hong-Gui Han*, Li-Dan Wang, Jun-Fei Qiao

***Abstract*—An approach, named extended extreme learning machine (ELM), is proposed for training the weights of a class of hierarchical feedforward neural network (HFNN). Unlike conventional single-hidden-layer feedforward networks (SLFNs), this hierarchical ELM (HELM) is based on the hierarchical structure which is capable of hierarchical learning of sequential information online, and one may simply choose hidden layers and then only need to adjust the output weights linking the hidden layer and the output layer. In such HELM implementations, the extended ELM provides better generalization performance during the learning process. Moreover, the proposed extended ELM method is efficient not only for HFNNs with sigmoid hidden nodes but also for HFNNs with radial basis function (RBF) hidden nodes. Finally, the HELM is applied to the activated sludge wastewater treatment processes (WWTPs) for predicting the water qualities. Experimental results and the performance comparison demonstrate the effectiveness of the proposed HELM.**

***Index Terms*—Hierarchical extreme learning machine; feedforward neural network; wastewater treatment process; predicting water qualities.**

I. INTRODUCTION

THE feedforward neural networks (FNNs) have been an active research topic in the recent years and they have been proposed as an efficient technique for various fields due to their universal approximation [1]-[3]. Numerous applications can be found in many disciplines; see [4]-[7], for some examples.

Out of many kinds of FNNs, single-hidden-layer feedforward networks (SLFNs) have been investigated as the most popular neural networks [8]. A SLFN consists of one input layer receiving the stimuli from external environments, one hidden layer, and one output layer sending the network output to external environments. In general, there are two main SLFN network architectures: the SLFNs with additive hidden nodes and radial basis function (RBF) nodes in the hidden layer [9]. The conventional trainings of SLFNs are mainly based on optimization theory. For example, a cost function is defined, which may be the mean squared error (MSE) between the output of a SLFN and the desired signal. The cost function is then minimized in the weight space, and a set of optimal weights will be obtained. In order for searching the optimal weights for SLFNs, a number of algorithms have been developed [10]-[11]. The gradient-based backpropagation (BP) and the recursive least squares (RLS) training algorithm are probably the most popular ones for SLFNs [12]-[13]. It is well known that the BP training algorithms

may have a slow convergence, and the searching for the global minimum point of a cost function may be trapped at local minima during gradient descent [14]. Moreover, if a network has large bounded input disturbances, the global minimum point may not be found. Therefore, the fast error convergence and strong robustness of the neural network with the BP algorithms may not be guaranteed. Compared to the BP algorithms, the RLS algorithms have a faster convergence speed. However, the RLS algorithms involve more complicated mathematical operations and require more computational resources than the BP algorithms [15].

To avoid the aforementioned problems, a novel learning algorithm – extreme learning machine (ELM) proposed by Huang *et al.* shows as a useful learning method to train SLFNs [16]. The ELM randomly generates hidden nodes parameters and then determines the output weights analytically. In addition, the ELM aims to reach not only the smallest training error but also the smallest norm of output weights [17]. It was established that ELM is an extremely fast batch learning algorithm and can provide good generalization performance [18], [40]. The key advantages of ELM (comparing with BP and RLS) are that ELM needs no iteration when determining the parameters, which dramatically reduces the computational time for training the SLFNs. In ELM, the linear equations are computed, the elements of the matrix should be inverted, and the output layer is linear. There are more data vectors than nodes in the hidden layer in practice. However, due to the ELM sometimes makes the hidden layer output matrix of SLFN not full column rank, Wang *et al.* proposes an improved algorithm for ELM [19]. This improved ELM makes a proper selection of the input weights and bias before calculating the output weights, which ensures the full column rank of the hidden layer output matrix in theory and extends the learning rate and the robustness property of SLFNs. Recently, to suitable for the Bayesian neural network models, a Bayesian ELM (BELM) is introduced by Emilio *et al.* [20]. The BELM presents some advantages over other approaches: it allows the introduction of *a priori* knowledge; obtains the confidence intervals without the need of applying methods that are computationally intensive; and presents high generalization capabilities. In fact, it is found that the BELM generally requires more hidden nodes than other conventional algorithms [21]. To address this problem, Zhu *et al.* propose an evolutionary extreme learning machine (E-ELM), in which the input weights and hidden layer biases are determined by using the differential evolution algorithm, and the output weights are determined by using the Moore-Penrose generalized inverse [22]. Additionally, Huynh *et al.* propose a regularized least-squares extreme learning machine (RLS-ELM) which can determine the input weights and biases of hidden nodes based on a fast regularized least-squares scheme [23]. Although ELM has been attracting the attentions from more and more researchers [24]-[26], [41]-[43], the ELM can only be used for the SLFNs with the single FNN structure. For example, if the structure of FNN is hierarchical (shown in Figure 1), the former ELM training algorithms are not suitable.

In fact, it has been demonstrated that the performance of FNNs is particularly relative to the qualities of the network input

variables [27]. Unfortunately, however, for some uncertain nonlinear complex systems, no obvious and *a priori* specified connection exists between the known input variables and the outputs. Moreover, in most of the nonlinear complex systems there are typically unknown input variables [28]. In most existing results of FNN design, to simplify the problem, the global system is often divided into several different subsystems and it is separately approximated in each subsystem [29]. However, when the key variables among each subsystems are uncertain, e.g., there are uncertain parameters and unknown nonlinear functions, the subsystems analysis becomes much more complex [30]. Moreover, multiple subsystems identifying may give each subsystem a higher resolution, but it neglects the cross-subsystem correlations and may result in some false results. To handle this problem, hierarchical structures should be included into a FNN [31].

In this paper, a hierarchical extreme learning machine (HELM) is proposed to improve the approximation ability of feedforward neural network for predicting the water qualities in the wastewater treatment processes (WWTPs). This algorithm has several advantages: firstly, in this HELM, the common part is the similar variable correlations over systems, and the specific part is the correlations that are not shared by all systems. Then the variables among uncertain nonlinear complex systems have been predicted. Secondly, the extended ELM is proposed for training the hierarchical FNN (HFNN). The extended ELM can retain good generalization performance for training the model. Enhancing the training algorithm itself leads to design well-behaved hierarchical FNN. Thirdly, the HELM is particularly focused on the modeling ability for the WWTPs. By analyzing both the common part and the specific part, this HELM can identify the different operating parameters and can effectively model the nonlinear complex system. Finally, the proposed algorithm is applied to predicting the water qualities in WWTPs, which can provide a basis for water treatment plant management decisions.

The rest of this paper is organized as follows. The next section briefly discusses the HFNN. In addition, the input-output selection method is discussed. In section III, the extended ELM is described to train the HFNN. In addition, the HELM is presented. The experimental results of the simulations on the real WWTPs are presented in Section IV. The performance of the HELM is compared with that of several other methods. The results demonstrate that the proposed HELM is efficient. Finally, conclusion shows in the end of the paper. For the sake of discussion, the acronyms used in this paper are list in the appendix.

II. Hierarchical Feedforward Neural Network (HFNN)

Without loss of generality, a single hidden layer HFNN is used in this paper. Figure 1 shows the structure of a HFNN. A HFNN has a network structure in terms of the direction of information flow. To better illustrate the HELM method described in this paper, the HFNN chosen is a two-hierarchical model. The structure of a two-hierarchical FNN consists of two parts. There are one input layer, one output layer and one hidden layer in every part. A mathematical description of each part in the HFNN is given below.

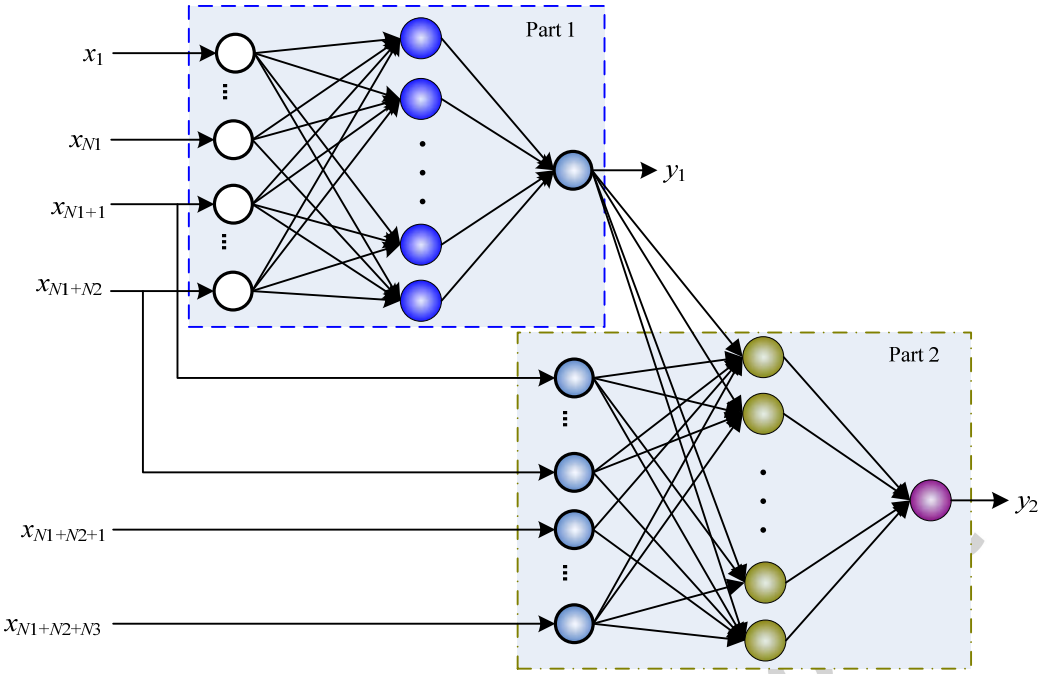


Fig.1 The hierarchical feedforward neural network (HFNN)

Part 1:

Input layer: there are $N1+N2$ nodes in this layer which represent the input variables of the first part. The output values of this input layer can be expressed as

$$u_i^1 = x_i, \quad i = 1, 2, \cdots, N1+N2, \tag{1}$$

where u_i^1 is the ith output value, and the input vector is given by $\mathbf{X}_1 = [\, x_1, x_2, \ldots, x_{N1+N2}\,]$.

Hidden layer: there are $M1$ nodes in this layer; each node connects with both the network's input and output nodes in the first part. The outputs are

$$\varphi_j^1 = f\Big(\sum_{i=1}^{N1+N2} w_{ij}^1 u_i^1\Big), \quad j = 1, 2, \cdots, M1, \tag{2}$$

where $f(x) = (1+e^{-x})^{-1}$, φ_j^1 is the output of the jth node, $\boldsymbol{\varphi}^1 = [\varphi_1^1, \varphi_2^1, \cdots, \varphi_{M1}^1]^{\mathrm{T}}$ are the output of the hidden layer whose dimension is $M1\times 1$, and w_{ij}^1 is the weight connecting the ith input node with the jth hidden node in the first part, $\mathbf{W}^1 = [\mathbf{w}_{\bullet 1}^1, \mathbf{w}_{\bullet 2}^1, \cdots, \mathbf{w}_{\bullet M1}^1]$ whose dimension is $M1\times(N1+N2)$, and $\mathbf{w}_{\bullet j}^1 = [w_{1j}^1, w_{2j}^1, \cdots, w_{M1j}^1]^{\mathrm{T}}$.

Output layer: there is one node in this layer; the output value is given by

$$y_1 = \sum_{j=1}^{M1} v_j^1 \varphi_j^1, \quad j = 1, 2, \cdots, M1, \tag{3}$$

where y_1 is the output of the first part, v_j^1 is the weight connecting the jth node in the hidden layer with the output node, $\mathbf{v}^1 = [v_1^1, v_2^1, \cdots, v_{M1}^1]$ whose dimension is $M1\times 1$.

***Remark 1**:* In order to simplify the discussion, the description is stated for a multi-input and single-output (MISO) network structure for example, the results could also be suitable for the multi-input and multi-output (MIMO) structure.

Part 2:

Input layer: the output values of input layer in the second part are given as

$$u_k^2 = x_k, \quad k = N1+1, \cdots, N1+N2, \cdots, N1+N2+N3, N1+N2+N3+1, \tag{4}$$

where u_k^2 is the kth output value, there are $N2+N3+1$ nodes in this layer, and the input vector is given by $\mathbf{X}_2 = [x_{N1+1}, x_{N1+2}, \ldots, x_{N1+N2+N3}, x_{N1+N2+N3+1}]$, and $x_{N1+N2+N3+1} = y_1$.

Hidden layer: there are $M2$ nodes in this layer; and the output values can be described by

$$\varphi_l^2 = f(\sum_{k=1}^{N2+N3} w_{kl}^2 u_k^2), \quad l = 1, 2, \cdots, M2, \tag{5}$$

where φ_l^2 is the output of the lth node, $\boldsymbol{\varphi}^2 = [\varphi_1^2, \varphi_2^2, \cdots, \varphi_{M2}^2]^{\mathbf{T}}$ are the output of the hidden layer whose dimension is $M2\times1$, and w_{kl}^2 is the weight value connecting the kth input node with the lth node in the hidden layer, $\mathbf{W}^2 = [\mathbf{w}_{\bullet 1}^2, \mathbf{w}_{\bullet 2}^2, \cdots, \mathbf{w}_{\bullet M2}^2]$ whose dimension is $M2\times(N2+N3)$, and $\mathbf{w}_{\bullet l}^2 = [w_{1l}^2, w_{2l}^2, \cdots, w_{M2\,l}^2]^{\mathbf{T}}$.

Output layer: the output value of the second part is given by

$$y_2 = \sum_{k=1}^{M2} v_k^2 \varphi_k^2, \tag{6}$$

where y_2 is the output value, v_k^2 is the weight connecting the kth hidden node with the output node in this second part, $\mathbf{v}^2 = [v_1^2, v_2^2, \cdots, v_{M2}^2]$.

***Remark 2**:* It should be pointed out that the idea of identifying unknown variable y_1 is proposed in the HFNN, and has been applied to obtain the more accuracy y_2 result for predicting key parameters of the uncertain nonlinear complex systems. This HFNN plays an important role in system and control areas. It has been shown that the HFNN gives a framework for predicting the unknown parameters using an input-output description based on the hierarchical structures design.

III. HIERARCHICAL EXTREME LEARNING MACHINE (HELM)

To train the HFNN automatically an extended ELM strategy is used in this paper. This training strategy adjusts the output weights of the HFNN by extending the ELM algorithm in the training process. The performance of the HELM with respect to the weights adjusting is an important issue and needs careful investigation. This is crucial for the successful applications.

It is clear that the success of the HFNN is highly dependent on the extended ELM. The extended ELM must effectively training the output weights of the HFNN, and also be easily usable in nonlinear systems. In this section, the extended ELM used

for the HFNN is analyzed. And the performance of the training algorithm is considered. Furthermore, through this analysis one obtains a better understanding on the HELM adjusting algorithms.

A. Extended Extreme Learning Machine

A multi-output HFNN with two-hierarchical (shown in Fig. 1) can be also described by

$$\mathbf{y}(p) = \mathbf{v}(p)\boldsymbol{\varphi}(p), \tag{7}$$

where

$$\begin{aligned} \mathbf{y}(p) &= [y_1(p), y_2(p)]^{\mathrm{T}}, \\ \mathbf{v}(p) &= [\mathbf{v}^1(p), \mathbf{v}^2(p)]^{\mathrm{T}}, \\ \boldsymbol{\varphi}(p) &= [\boldsymbol{\varphi}^1(p), \boldsymbol{\varphi}^2(p)], \end{aligned} \tag{8}$$

and $p = 1, 2, \ldots, P$, there are P arbitrary distinct samples $\{\mathbf{X}(p), \mathbf{t}(p) | p = 1, 2, \ldots, P\}$, $\mathbf{X}(p)=[x_1(p), x_2(p), \ldots, x_{N1+N2+N3}(p)]$ is the input vector of the HFNN, $\mathbf{t}(p)=[t_1(p), t_2(p)]$ is the desired output value, $\mathbf{y}(p)$ is the output values, $\mathbf{v}(p)$ is the output weights and $\boldsymbol{\varphi}(p)$ is the output value of the hidden layer for the pth sample.

This HFNN with $M1+M2$ hidden nodes ($M1$ for the first part and $M2$ for the second part) with activation function $f(x)$ can approximate these P samples with zero error means that

$$\boldsymbol{\theta}\mathbf{V} = \mathbf{T}, \tag{9}$$

where $\mathbf{T} = [\mathbf{t}(1), \mathbf{t}(2), \ldots, \mathbf{t}(P)]$ is the desired output vector, $\boldsymbol{\theta}= [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \ldots, \boldsymbol{\varphi}(P)]$ is the output of the hidden layer, and $\mathbf{V} = [\mathbf{v}(1), \mathbf{v}(2), \ldots, \mathbf{v}(P)]^{\mathrm{T}}$ is the output weight vector.

Due to the first output variable y_1 is the input variable in the second part; this zero error formation can be represented as the following equations

$$\begin{cases} \boldsymbol{\theta}^1\mathbf{V}^1 = \mathbf{T}^1, \\ \boldsymbol{\theta}^2\mathbf{V}^2 = \mathbf{T}^2, \end{cases} \tag{10}$$

where $\mathbf{T}^1 = [t_1(1), t_1(2), \ldots, t_1(P)]$, $\boldsymbol{\theta}^1= [\boldsymbol{\varphi}^1(1), \boldsymbol{\varphi}^1(2), \ldots, \boldsymbol{\varphi}^1(P)]$ and $\mathbf{V}^1 = [\mathbf{v}^1(1), \mathbf{v}^1(2), \ldots, \mathbf{v}^1(P)]^{\mathrm{T}}$ are the desired output vector, the hidden layer output vector and the output weight vector in the first part, respectively. And $\mathbf{T}^2$, $\boldsymbol{\theta}^2$ and $\mathbf{V}^2$ are the desired output vector, the hidden layer output vector and the output weight vector in the second part.

The difference between the new proposed extended ELM algorithm and the ELM algorithm lies in the training of output weights. Then, the smallest norm least squares solution of the above (10) is

$$\mathbf{V}^1 = (\boldsymbol{\theta}^1)^*\mathbf{T}^1, \tag{11}$$

and

$$\mathbf{V}^2 = (\boldsymbol{\theta}^2)^*\mathbf{T}^2, \tag{12}$$

where $(\boldsymbol{\theta}^1)^*$ and $(\boldsymbol{\theta}^2)^*$ is the Moore-Penrose generalized inverse of matrix $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$.

Nevertheless, the random selection sometimes produces nonsingular hidden layer output matrix which causes no solution of the equations (11) and (12). To overcome the shortcoming, an extra regularizing term λ, where λ is a small constant and $\mathbf{I}$ unit matrix, should be added to the solution for guaranteeing that the matrix to be inverted is nonsingular to keep $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$ full column ranks

$$(\boldsymbol{\theta}^1)^* = (\lambda \mathbf{I} + (\boldsymbol{\theta}^1)^{\mathbf{T}} \boldsymbol{\theta}^1)^{-1} (\boldsymbol{\theta}^1)^{\mathbf{T}}, \tag{13}$$

and

$$(\boldsymbol{\theta}^2)^* = (\lambda \mathbf{I} + (\boldsymbol{\theta}^2)^{\mathbf{T}} \boldsymbol{\theta}^2)^{-1} (\boldsymbol{\theta}^2)^{\mathbf{T}}, \tag{14}$$

where λ is a positive value which is obtained as [44]. This extra method can be generally used to calculate the Moore-Penrose generalized inverse of $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$ in all cases.

The extended ELM algorithm consists of two main phases. The first phase is to train HFNN using the ELM algorithm with some batch of training data for part 1. After the first phase, the extended ELM algorithm will train the output weights of the second part, and then, all the output weights will be adjusted. For this, the extended ELM algorithm is useful for training HFNN. So the extended ELM is actually more effective than ELM. And the extended ELM algorithm also expands the use of the ELM algorithm including easy implementing, good generalization performance. Moreover, the new algorithm improved the effectiveness of learning: the full column rank property of the matrix makes the orthogonal projection method available.

Remark 3*:* One of the typical implementations of extended ELM is to train the FNN with hierarchical structures – HFNN. It is interesting to see that the extended ELM can be used to the systems with hierarchical variables connections exist between the inputs and the outputs.

B. Hierarchical Extreme Learning Machine (HELM)

A HELM has a hierarchical network structure in terms of the direction of information flow. The parameter learning algorithm - extended ELM is focused on fast and effective methods that can be used to train the output weights of the HFNN. The main steps in the proposed HELM can be summarized as follows.

Step 1) Create an initial two-hierarchical HFNN. The number of nodes in the input and output layers is the same as the number of input and output variables in the problem that is being solved. The number of nodes in the hidden layer is $M1$ in the first part and is $M2$ in the second part. All of the parameters should be initialized with the connection weights of the HFNN uniformly distributed within a small range.

Step 2) For the initial set of input sample $\{\mathbf{X}(p), \mathbf{t}(p) | p = 1, 2, \ldots, P_0\}$, calculate the hidden layer output matrix $\boldsymbol{\theta}_0^1$ of the HFNN in the first part, and adjust the first part output weights $\mathbf{V}_0^1$ using the training rules (11) and (13)

$$\mathbf{V}_0^1 = \mathbf{Q}_0^1(\boldsymbol{\theta}_0^1)^{\mathrm{T}}\mathbf{T}_0^1, \tag{15}$$

where $\mathbf{Q}_0^1 = (\lambda\mathbf{I} + (\boldsymbol{\theta}_0^1)^{\mathrm{T}}\boldsymbol{\theta}_0^1)^{-1}$, and $\mathbf{T}_0^1 = [t_1(1), t_1(2), \cdots, t_1(P_0)]^{\mathrm{T}}$. Then, calculate the hidden layer output matrix $\boldsymbol{\theta}_0^2$ of the HFNN in the second part, and estimate the second part output weights $\mathbf{V}_0^2$ using the training rules (12) and (14)

$$\mathbf{V}_0^2 = \mathbf{Q}_0^2(\boldsymbol{\theta}_0^2)^{\mathrm{T}}\mathbf{T}_0^2, \tag{16}$$

where $\mathbf{Q}_0^2 = (\lambda\mathbf{I} + (\boldsymbol{\theta}_0^2)^{\mathrm{T}}\boldsymbol{\theta}_0^2)^{-1}$, and $\mathbf{T}_0^2 = [t_2(1), t_2(2), \cdots, t_2(P_0)]^{\mathrm{T}}$. Set $a = 0$.

Step 3) Present the (a+1)th set of new samples $\{\mathbf{X}(p), \mathbf{t}(p) | p = P_a+1, P_a+2, \ldots, P_a+P_{a+1}\}$, P_{a+1} is the number of new samples. Calculate the hidden layer output matrix $\boldsymbol{\theta}_{a+1}^1$ of the HFNN in the first part using the training rules (11), and adjust the first part output weights $\mathbf{V}_{a+1}^1$ as in [18]

$$\begin{aligned}\mathbf{Q}_{a+1}^1 &= \mathbf{Q}_a^1 - \mathbf{Q}_a^1(\boldsymbol{\theta}_{a+1}^1)^{\mathrm{T}}(\mathbf{I} + \boldsymbol{\theta}_{a+1}^1\mathbf{Q}_a^1(\boldsymbol{\theta}_{a+1}^1)^{\mathrm{T}})^{-1}\boldsymbol{\theta}_{a+1}^1\mathbf{Q}_a^1,\\ \mathbf{V}_{a+1}^1 &= \mathbf{V}_a^1 + \mathbf{Q}_{a+1}^1(\boldsymbol{\theta}_{a+1}^1)^{\mathrm{T}}(\mathbf{T}_{a+1}^1 - \boldsymbol{\theta}_{a+1}^1\mathbf{V}_a^1),\end{aligned} \tag{17}$$

and then calculate the hidden layer output matrix $\boldsymbol{\theta}_{a+1}^2$ of the HFNN in the second part using the training rules (12), and estimate the second part output weights $\mathbf{V}_{a+1}^2$

$$\begin{aligned}\mathbf{Q}_{a+1}^2 &= \mathbf{Q}_a^2 - \mathbf{Q}_a^2(\boldsymbol{\theta}_{a+1}^2)^{\mathrm{T}}(\mathbf{I} + \boldsymbol{\theta}_{a+1}^2\mathbf{Q}_a^2(\boldsymbol{\theta}_{a+1}^2)^{\mathrm{T}})^{-1}\boldsymbol{\theta}_{a+1}^2\mathbf{Q}_a^2,\\ \mathbf{V}_{a+1}^2 &= \mathbf{V}_a^2 + \mathbf{Q}_{a+1}^2(\boldsymbol{\theta}_{a+1}^2)^{\mathrm{T}}(\mathbf{T}_{a+1}^2 - \boldsymbol{\theta}_{a+1}^2\mathbf{V}_a^2),\end{aligned} \tag{18}$$

where $\mathbf{T}_{a+1}^1 = [t_1(P_a+1), t_1(P_a+2), \cdots, t_1(P_{a+1})]^{\mathrm{T}}$. and $\mathbf{T}_{a+1}^2 = [t_2(P_a+1), t_2(P_a+2), \cdots, t_2(P_{a+1})]^{\mathrm{T}}$.

Step 4) a=a+1, go to Step 3. Stop when a=$\bar{a}$, $\bar{a}$ is the number of the training samples' sets.

Remark 4*:* In fact, the fundamentals of HELM are composed of twofold: universal approximation capability with easy and fast implementations, and hierarchical structure for indirect connections between the inputs and the outputs. Furthermore, HELM is an online sequential learning algorithm which is potentially important to model nonlinear systems. HELM can learn the train data one by one or set by set, and then, all the training data will be discarded once the learning procedure on these data is completed. HELM consists of a few of hierarchical structures that may have different adaptabilities to the indirect variables.

IV. Experimental Studies

The performance of HELM is verified by applying it to WWTPs. The performance of the method was evaluated by comparing the results with other methods. All the simulations are programmed with Matlab version 7.01, and were run on a Pentium 4 with a clock speed of 2.6 GHz and 1 GB of RAM, under a Microsoft Windows XP environment.

In the following simulations the learning parameters for all the algorithms are set independently so that each network solution will, on average, obtain optimal performance. Moreover, the performance of the methods is measured using the root mean-square-error (RMSE) function which is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{p=1}^{n}(y(p)-t(p))^2}, \tag{19}$$

where n is the number of the training samples, $y(p)$ and $t(p)$ are the pth calculated output and the desired output respectively.

A. Scheme Setup

Predicting water qualities in WWTPs can provide a basis for water treatment plant management decisions that can minimize microbial risk and optimize the treatment operation [32]. Also, a poor performance of a predictor can destroy the result of the best control method. In many practical situations, however, it is difficult to predict accurately the qualities of the water in the treatment process due to a lack of knowledge of the parameters used in the process, or the presence of disturbances in the system. Thus, the predictor should take an appropriate action to counteract the presence of disturbances to which the system is subjected and should be able to adjust itself to the changing dynamics of the system.

In this section, the main objective is to develop a water quality prediction model that provides accurate predictions of the biochemical oxygen demand (BOD) and sludge volume index (SVI) in the WWTPs using the proposed HELM. In this experiment, the most important water qualities were selected as the input research variables. BOD is related to the parameters such as influent flow (Q_{in}), dissolved oxygen (DO) concentration, pH, mixed liquor suspended solid (MLSS) and total nutrients (TN) [33]. And SVI is related to the parameters such as Q_{in}, DO concentration, pH, TN, chemical oxygen demand (COD) and BOD [34]. However, for SVI, BOD is an unknown input variable. Since an approach based on the HELM does not make any assumptions about the functional relationship between the dependent and independent variables, it is suitable for capturing functional relationships between bacterial levels and other variables. In addition, the HELM can utilize the predicting BOD values to predict SVI. For above reasons, the HELM is used to predict both BOD and SVI values in WWTPs.

Table 1. Inputs used in this study

Variables	*Explanation*	*BOD*	*SVI*
Q_{in} (m^3/d)	Influent flow	√	√
DO (mg/l)	Dissolved oxygen concentration	√	√
pH	Acidity and basicity	√	√
MLSS (mg/l)	Mixed liquor suspended solid	√	-
COD (mg/l)	Chemical oxygen demand	-	√
TN (mg/l)	Total nutrients	√	√
BOD (mg/l)	Biochemical oxygen demand	-	√

√ means the variable is considered as the input.
- means the variable is not considered as the input.

The data used as inputs for the HELM included the measurements routinely performed at the WWTPs. All data are collected on a daily basis and covered all four seasons. They are either come from routine input by lab technicians or related experimental data during special campaigns. All data are not collected at a regular rate and are often missing during weekends. Missing values are replaced by linearly interpolated values. The daily frequency of measurements is considered sufficient because of the long residence times in WWTPs. The inputs considered are listed in Table 1. All variables are normalized and de-normalized between 0 and 1 before and after application in the neural network.

B. HELM Predicting and Evaluation

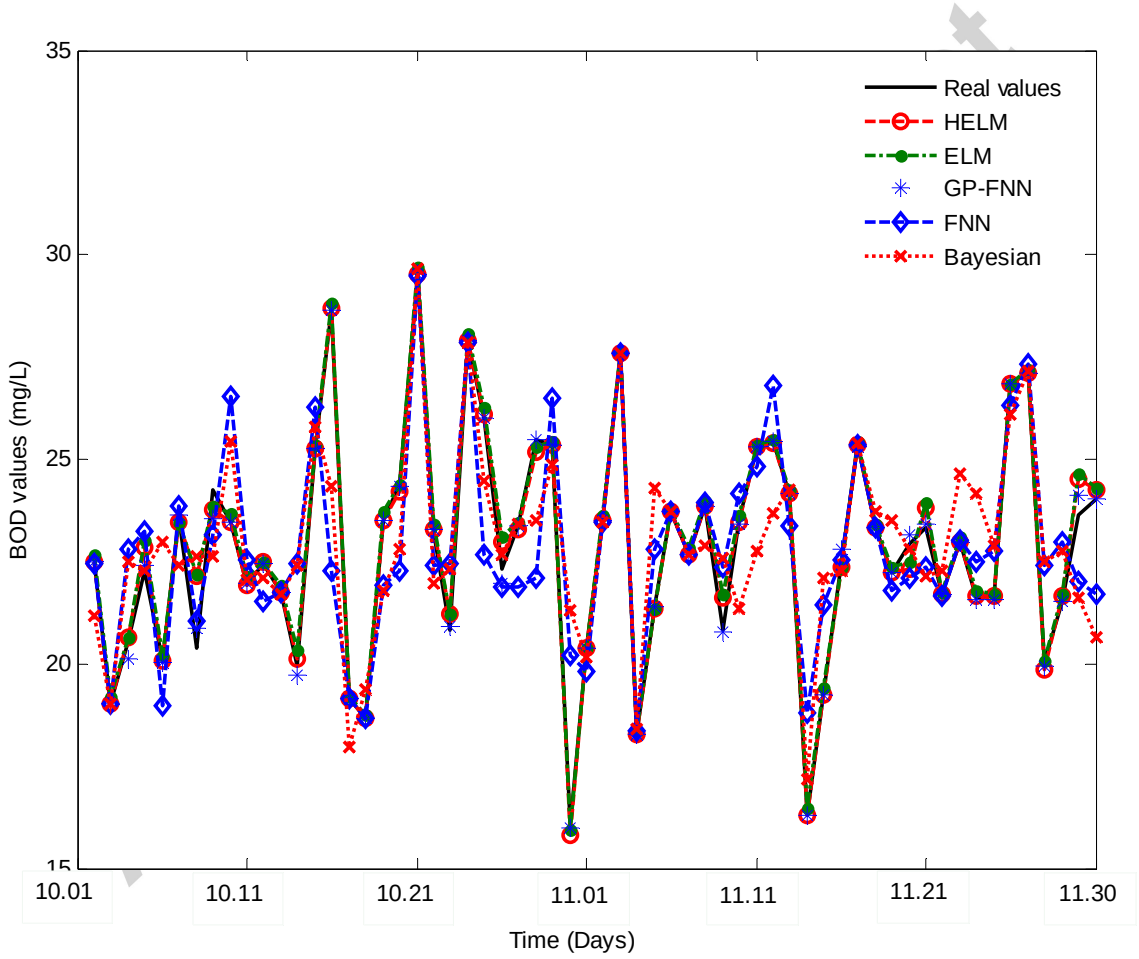


Fig.2. BOD values (Case 1)

In this paper, the HELM is proposed to predict the BOD and SVI values: its aim to own the BOD and SVI values on-line based on the related parameters. The input–output water quality data from the real WWTPs (Beijing, China) were measured over the year 2010. After deleting abnormal data, 360 samples were obtained and normalized; sixty samples from October and November were used as testing data whilst the remaining 300 samples were employed as training data. The error measures for

the BOD and SVI are 3mg/L confidence limits. In this work, two aspects of the evaluations are discussed: BOD predicting evaluation (case 1) and SVI predicting evaluation (case 2).

(1) Case 1 study

The predicting values are compared with the Bayesian approach [35], the FNN [36], the ELM [23] and the growing and pruning fuzzy neural network (GP-FNN) [37].

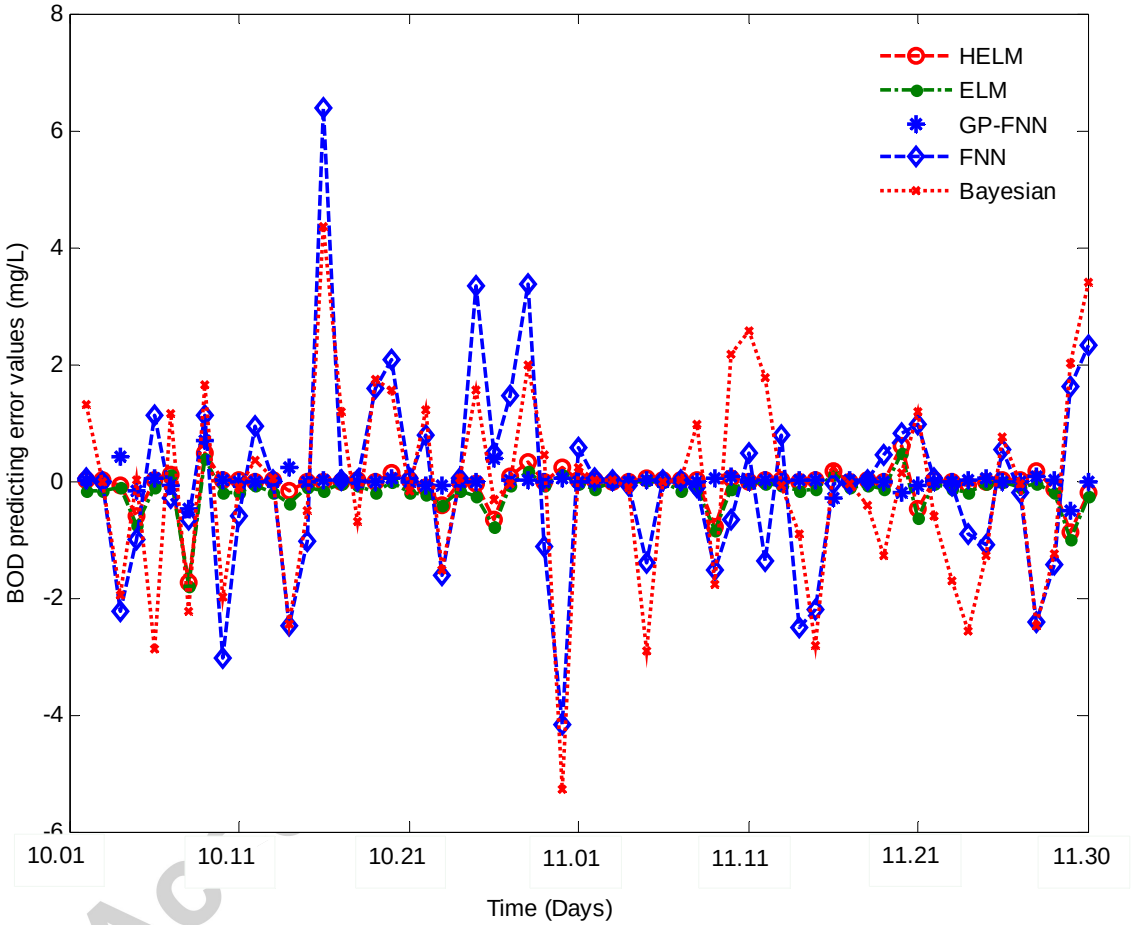


Fig.3. The BOD error between the plant output and the predicting values (Case 1)

In this case, first, 150 and 180 initial hidden nodes are used for the first part and the second part of the proposed HELM, respectively. The number of hidden nodes is chosen 150 in the ELM for this case. The parameters of the Bayesian approach, FNN and GP-FNN are the same as the initial papers. In the experiments, the BOD values by the adaptive methods are depicted together with the real process output in Figure 2 for comparison. In order to check the predicting abilities of the different methods, the errors of the different methods are shown in Figure 3. Clearly, the predicting values obtained using the proposed HELM model with respect to the real output is more accurate than those obtained from the Bayesian and the FNN models.

Performance is assessed by the RMSE as equation (19) and the predicting accuracy ($\sum_{p=1}^{Days}(1-e(p)/t(p))/Days$). The details are presented in Table 2.

The comparison results in Figures 2-3 demonstrate the superiority of the HELM in predicting BOD values for the real WWTPs. Figure 3 shows that the HELM can achieve better performance than the Bayesian approach, the FNN and the ELM methods for predicting BOD. However, the performance of the GP-FNN is better than the proposed HELM method.

Table 2 shows the different methods for predicting the BOD values. In this table the average accuracy (testing results) and the RMSE are shown. The FNN, the GP-FNN, the ELM and the proposed HELM model are self-adaptive. The training RMSE of the HELM model is the smallest. And the accuracy of the HELM is better than that of the Bayesian approach, the FNN and the ELM methods. The comparisons demonstrate that the HELM model is suitable for the BOD prediction.

Table 2. A comparison of the performance of different models (all results were averaged on 50 independent runs, Case 1)

Methods	*Hidden nodes*	*Training RMSE*	*Testing Accuracy (%)*	
			Min	*Mean*
HELM	**150 (First part)**	**0.074**	**96.20**	**98.01**
ELM	150	**0.082**	94.41	96.64
GP-FNN	9*	0.551*	97.19*	98.81*
FNN	50*	0.905*	93.48*	95.52*
Bayesian approach	/	0.918*	90.15*	94.32*

*The results are listed in original papers.
/ There is no meaning for hidden neurons (Bayesian Approach is not a network).

(2) Case 2 study

In this case, the predicting performance of the HELM model is compared to that in the other SVI models: the dynamic ARX method [38], the FNN [39], the GP-FNN and the ELM.

Table 3. A comparison of the performance of different models (all results were averaged on 80 independent runs, case 2)

Methods	*Hidden nodes*	*Training RMSE*	*Testing Accuracy (%)*	
			Min	*Mean*
HELM	**180 (Second part)**	**0.084**	**96.01**	**97.71**
ELM	180	0.177	87.12	87.78
GP-FNN	11	0.851	88.09	90.11
FNN	50*	1.101*	81.10*	88.50*
Dynamic ARX	/	/	79.10*	85.12*

To show a fair comparison between the HELM and the other methods, the input variables used for the FNN, the GP-FNN and the ELM are Q_{in}, DO concentration, pH, TN and COD. However, due to novel structure of the proposed HELM, one more input variable – BOD is used to predict the SVI values. The other parameters of the models are the same as the initial papers. The number of hidden nodes is chosen 180 in the ELM for this case.

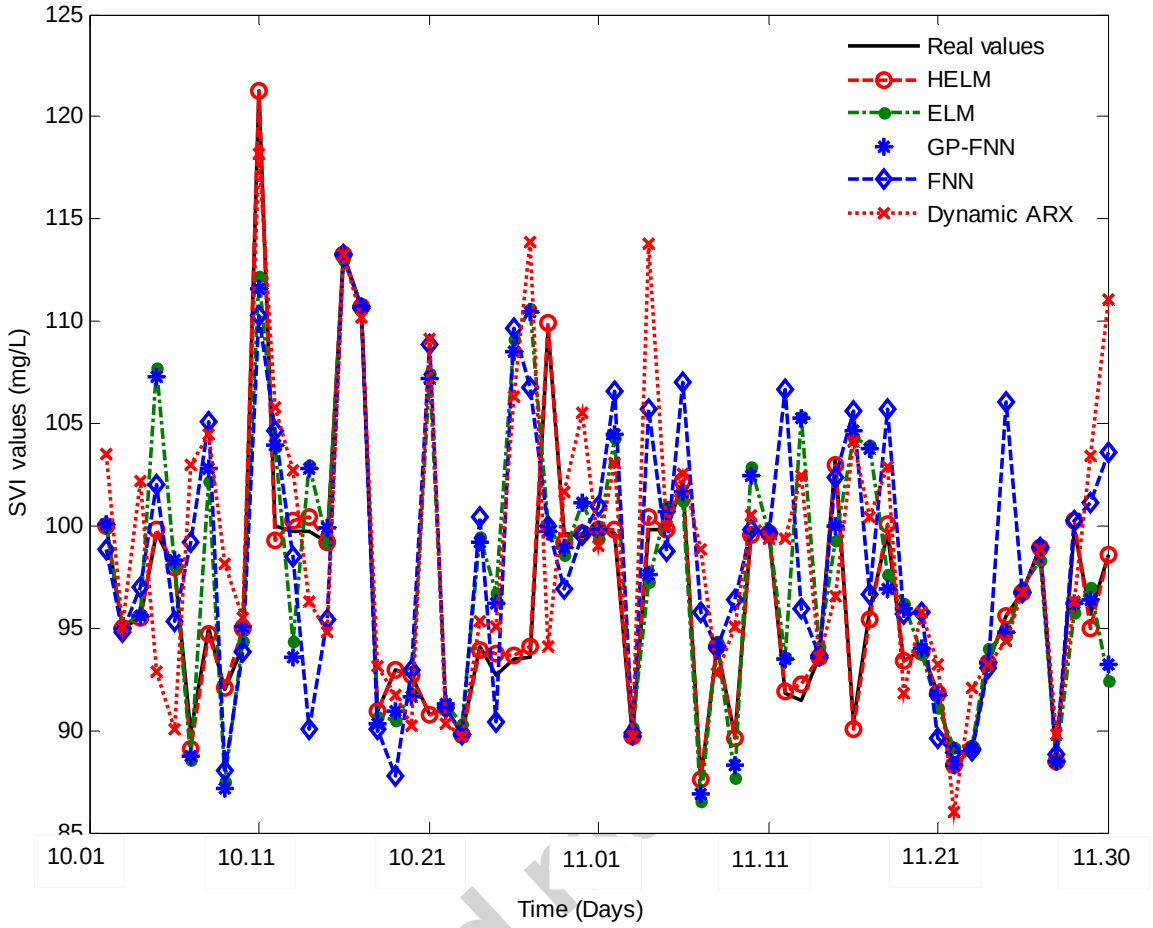


Fig.4. SVI values (Case 2)

In this case, the SVI values by the different methods are displayed together in Figure 4. The errors of the different methods are shown in Figure 5. The performance is compared in Table 3. By analyzing Tables 3, the HELM has the following advantages for SVI predicting compared to other methods.

1) The HELM obtains the best training RMSE values for the SVI prediction in this case.

2) The HELM gets the best accuracy both for the minima and mean values as shown in Table 3. The results show that the SVI values can be predicted well by the HELM to meet the limits specified by the regulations.

3) This hierarchical characteristic of HELM is very useful to predict the SVI values for the WWTPs. The HELM can use BOD which can not be measured on-line as the input variable.

From the above experimental results on the BOD and SVI prediction, the proposed HELM is a suitable and effective method for predicting both the BOD and SVI values. It can also be seen that, in these two cases, the HELM is more accurate than the Bayesian approach, the FNN, the ELM and the GP-FNN in case 1. The HELM is the most accurate in case 2. The model is relatively straightforward to implement on-line and can be used to make real-time predictions of the BOD and SVI.

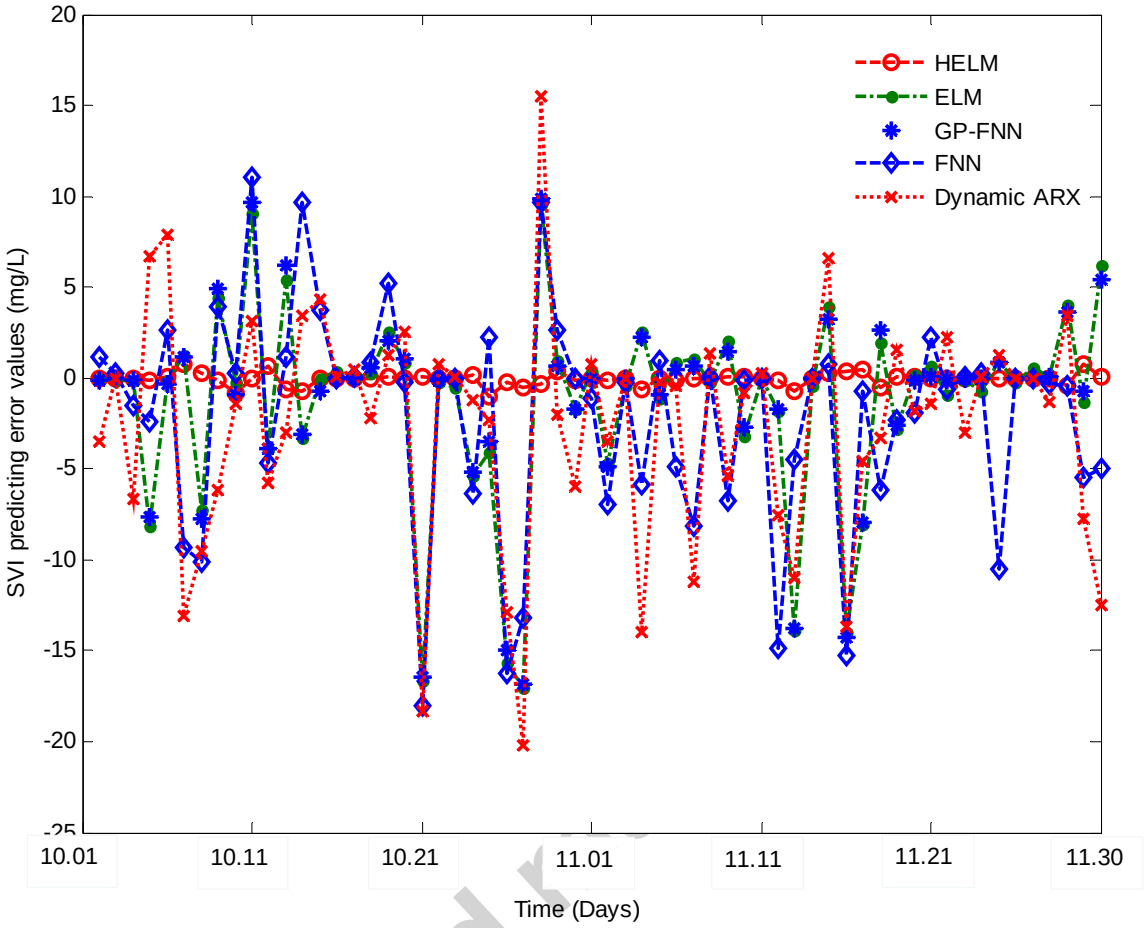


Fig.5. The SVI error between the plant output and the predicting values (Case 2)

C. Analysis of the Experimental Results

For comparison purposes, Bayesian approach, the FNN, the GP-FNN, and the ELM are also employed to the same experimental system to show the BOD predicting performance. Table 2 shows that the HELM method can obtain higher mean accuracy than that in the Bayesian approach, the FNN and the ELM in the BOD predicting. The results demonstrate that the BOD trends in WWTPs can be predicted with acceptable accuracy using the Q_{in}, DO, pH, MLSS and TN data as input variables.

To gain deeper understanding of the proposed HELM method, the special comparisons of the SVI predicting performance among the three methods – the dynamic ARX method, the FNN, the GP-FNN and the ELM – were analyzed. According to the results of the paper, it can be seen that the proposed HELM featuring higher predicting performance can be effectively applied to predict the SVI values in WWTPs. Predicting SVI by HELM can give accurate results (see Table 3). Performance is generally improved with increasing the input variable – BOD. The results demonstrate that the SVI trends in WWTPs can be predicted with acceptable accuracy using the Q_{in}, DO, pH, BOD, COD, and TN data as input variables.

V. Conclusion

This paper has introduced a hierarchical structure HELM for predicting the water qualities in WWTPs. This approach is different from traditional approaches because its learning algorithm actually addresses the characteristics of the system. The major advantage of the HELM method is that extended ELM and the hierarchical structure of the HELM keep the model attuned with the current system dynamics. This method enhances the capacity of the SLFNs to adapt to nonlinear dynamic systems. The adaptive model developed in this research is shown to yield more accurate predictions than the other methods in WWTPs context. Moreover, the proposed extended ELM method is efficient not only for HFNNs with sigmoid hidden nodes but also for HFNNs with RBF hidden nodes.

Moreover, the predicting performance shows that the HELM can match system nonlinear dynamics. Therefore, this predicting method performs well in the whole operating space. It is observed that the HELM can predict the BOD values, and then inspect the SVI values. Simulation proves that the HELM approach is of predicting the SVI values more accurately than the other models, which is essential to develop an efficient model prediction controller (MPC) according to the predicting results. Control strategies can be developed and implemented in a more proactive way.

Acknowledgment

The authors would like to thank Prof. Wen Yu for reading the manuscript and providing valuable comments. The authors also would like to thank the anonymous reviewers for their valuable comments and suggestions, which helped improve this paper greatly.

APPENDIX

Table 4: Lists of acronyms

Acronym	Description
ARX	Auto-regressive exogenous
BP	Backpropagation
BOD	Biological oxygen demand over a 5-day period
COD	Chemical oxygen demand
DO	Dissolved oxygen
ELM	Extreme learning machine
HELM	Hierarchical extreme learning machine
FNN	Feedforward neural network
HFNN	Hierarchical feedforward neural network
GP-FNN	Growing and pruning fuzzy neural network
MPC	Model prediction controller
MIMO	Multi-input and multi-output
MISO	Multi-input and single-output
MLSS	Mixed liquor suspended solid
MSE	Mean square error
RBF	Radial basis function
RMSE	Root mean-square-error
RLS	Recursive least squares
SLFNs	Single-hidden-layer feedforward networks
SVI	Sludge volume index
TN	Total nutrients
WWTPs	Wastewater treatment processes

REFERENCES

[1] S. Haykin, Neural networks and learning machines, 3rd ed., Pearson Prentice Hall, 2009.
[2] F. Ham and I. Kostanic, Principles of neurocomputing for science and engineering, McGraw-Hill, 2001.
[3] T. Chow and S. Y. Cho, Neural networks and computing - learning algorithms and applications, Imperial College Press, 2007.
[4] H. G. Han, J. F Qiao, Adaptive dissolved oxygen control based on dynamic structure neural network, *Applied Soft Computing* 11(4) (2011) 3812-3820.
[5] H. G. Han, Q.L. Chen, J. F Qiao, An efficient self-organizing RBF neural network for water quality predicting, *Neural Networks* 24(7) (2011) 717-725.
[6] Y. W. Zhang, T. Y Chai, Z. M. Li, C. Y. Yang, Modeling and monitoring of dynamic processes, *IEEE Transactions on Neural Networks and Learning Systems* 23(2) (2012) 277–284.
[7] http://users.abo.fi/abulsari/EANN.html.
[8] G. B. Huang, L. Chen, C. K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Transactions on Neural Networks* 17(4) (2006) 878–891.
[9] G. B. Huang, D. H. Wang, Y. Lan, Extreme learning machines: a survey, *International Journal of Machine Learning and Cybernetics* 2(2) (2011) 107–122.
[10] J. F Qiao, H. G. Han, A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modeling, *International Journal of Neural Systems* 20(1) (2010) 63–74.
[11] Z. H. Man, K. Lee, D. H. Wang, Z. W. Cao, and S. Y. Khoo, Robust single-hidden layer feedforward network-based pattern classifier, *IEEE Transactions on Neural Networks and Learning Systems* 23(12) (2012) 1974–1986.
[12] J. Sum, C. S. Leung, and K. Ho, Convergence analyses on on-line weight noise injection-based training algorithms for MLPs, *IEEE Transactions on Neural Networks and Learning Systems* 23(11) (2012) 1827–1840.
[13] J. Bilski, L. Rutkowski, A fast training algorithm for neural networks, *IEEE Transactions on Circuits Systems II*, 1998 45(6) 1580–1591.
[14] R. Zhang, Z. B. Xu, G. B. Huang, D. H. Wang, Global convergence of online BP training with dynamic learning rate, *IEEE Transactions on Neural Networks and Learning Systems* 23(2) (2012) 330–341.

[15] M. S. Al-Batah, N. A. M. Isa, K. Z. Zamli, K. A. Azizli, Modified recursive least squares algorithm to train the hybrid multilayered perceptron (HMLP) network, *Applied Soft Computing* 10(1) (2010) 236–244.
[16] G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70(1-3) (2006) 489–501.
[17] R. Zhang, Y. Lan, G. B. Huang, Z. B. Xu, Universal approximation of extreme learning machine with adaptive growth of hidden nodes, *IEEE Transactions on Neural Networks and Learning Systems* 23(2) (2012) 365–371.
[18] G. B. Huang, H. M. Zhou, X. J. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 42(2) (2012) 513–529.
[19] Y. G. Wang, F. L. Cao, Y. B. Yuan, A study on effectiveness of extreme learning machine, *Neurocomputing* 74(16) (2011) 2483–2490.
[20] S. O. Emilio, G. S. Juan, D. M. Jose, V. F. Joan, M. Marcelino, R. M. Jose, A. J. Serrano, BELM: Bayesian extreme learning machine, *IEEE Transactions on Neural Networks* 22(3) (2011) 505–509.
[21] Y. M Yang, Y. N. Wang, and X. F. Yuan, Bidirectional extreme learning machine for regression problem and its learning effectiveness, *IEEE Transactions on Neural Networks and Learning Systems* 23(9) (2012) 1498–1505.
[22] Q. Y. Zhu, A. K. Qin, P. N. Suganthan, G. B. Huang, Evolutionary extreme learning machine, Pattern Recognition 38(10) (2005) 1759–1763.
[23] H. T. Huynh, Y. Won, J. J. Kim, An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks, *International Journal of Neural Systems* 18(5) (2008) 433–441.
[24] G. Feng, G.B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, *IEEE Transactions on Neural Networks* 20(8) (2009) 1352–1357.
[25] Z. L. Sun, T. M. Choi, K. F. Au, Y. Yu, Sales forecasting using extreme learning machine with applications in fashion retailing, *Decision Support Systems* 46(1) (2008) 411–419.
[26] H. J. Rong, Y. S. Ong, A. H. Tan, Z. Zhu, A fast pruned extreme learning machine for classification problem, *Neurocomputing* 72(1-3) (2008)359–366.
[27] H. G. Han, J. F Qiao, Prediction of activated sludge bulking based on a self-organizing RBF neural network, *Journal of Process Control* 22(6) (2012) 1103-1112.
[28] A. K. Kostarigka, G, A. Rovithakis, Adaptive dynamic output feedback neural network control of uncertain MIMO nonlinear systems with prescribed performance, *IEEE Transactions on Neural Networks and Learning Systems* 23(1) (2012) 138–149.
[29] Y. N. Li, C. G. Yang, S. Z. S. Ge, T. H. Lee, Adaptive output feedback NN control of a class of discrete-time MIMO nonlinear systems with unknown control directions, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 41(2) (2012) 507–517.
[30] J. Mårtensson, H. Hjalmarsson, How to make bias and variance errors insensitive to system and model complexity in identification. *IEEE Transactions on Automatic Control* 56(1) (2011) 100–112.
[31] L. Cheng, Z. G. Hou, M. Tan, Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model, *Automatica* 45(10) (2009) 2312–2318.
[32] H. G. Han, J. F Qiao, Q. L. Chen, Model predictive control of dissolved oxygen concentration based on a self-organizing RBF neural network, *Control Engineering Practice* 20(4) (2012) 465–476.
[33] C. S. Akratos, J. N. E. Papaspyros, V. A. Tsihrintzis, An artificial neural network model and design equations for BOD and COD removal prediction in horizontal subsurface flow constructed wetlands, *Chemical Engineering Journal* 143(1-3) (2008) 96–110.
[34] S. M. Kotay, T. Datta, J. Choi, R. Goel, Biocontrol of biomass bulking caused by Haliscomenobacter hydrossis using a newly isolated lytic bacteriophage, *Water Research* 45(9) (2011) 694–704.
[35] Y. Liu, P. J. Yang, C. Hu, H. C. Guo, Water quality modeling for load reduction under uncertainty: A Bayesian approach, *Water Research* 42(13) (2008) 3305–3314.
[36] V. Chandramouli, G. Brion, T. R. Neelakantan, S. Lingireddy, Backfilling missing microbial concentrations in a riverine database using artificial neural networks, *Water Research* 41(1) (2008) 217–227.
[37] H. G. Han, J. F Qiao, A self-organizing fuzzy neural network based on growing-and-pruning algorithm, *IEEE Transactions on Fuzzy Systems* 18(6) (2010) 1129–1143.
[38] I. Y. Smets, E. N. Banadda, J. Deurinck, N. Renders, R . Jenne, J. F. V. Impe, Dynamic modeling of filamentous bulking in lab-scale activated sludge processes, *Journal of Process Control* 16(3) (2006) 313-319.
[39] J. M. Brault, R. L. Labib, M. Perrier, P. Stuart, Prediction of activated sludge filamentous bulking using ATP data and neural networks, *Canadian Society for Chemical Engineering* 89(2) (2010) 1-13.
[40] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization, *Neurocomputing* 74(16) (2011) 2413-2421.
[41] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse, GPU-accelerated and parallelized ELM ensembles for large-scale regression, *Neurocomputing* 74(16) (2011) 2430-2437.
[42] Y. Lan, Y. C. Soh, and G. B. Huang, Ensemble of online sequential extreme learning machine, *Neurocomputing* 72(13-15) (2009) 3391-3395.
[43] J. H. Zhai, H. Y. Xu, and X. Z. Wang, Dynamic ensemble extreme learning machine based on sample entropy, *Soft Computin*g 16(9) (2012) 1493-1502.
[44] C. Bishop, "Pattern recognition and machine learning," Springer, 2006.

Honggui Han received the B.S. degree from Civil Aviation University of China in 2005; and received the Ph.D. degree from Beijing University of Technology in 2011. He is work in modeling and control in complex process, pattern identification, intelligent system and so on. Communication address: College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China. E-mail: Rechardhan@sina.com.

Mr. Han is a member of the IEEE Computational Intelligence Society. He is currently a reviewer of Control Engineering Practice, IEEE Transactions on Neural Networks and Learning Systems, and IEEE Transactions on Fuzzy Systems.

Lidan Wang received the B.S. degree from Dezhou University in 2012; he is a M.S. student in Beijing University of Technology. Communication address: College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China. E-mail: Wanglidan01@163.com.

Junfei Qiao received the Ph.D. degree from Northeast University in 1998; he is a professor and Ph.D. tutor in Beijing University of Technology. He is a professor of modeling and control in complex process, intelligent computing, and intelligent optimization and so on. Communication address: College of Electronic and Control Engineering, Beijing University of Technology, Beijing, E-mail: isibox@sina.com.

Manuscript received_____ This work was supported by the National Science Foundation of China under Grants 61203099, 61225016, 61034008, and 61004051, Beijing Municipal Natural Science Foundation under Grant 4122006, HongKong Scholor under Grant XJ2013018, Beijing Nova program under Grant Z131104000413007, and Ph.D. Program Foundation from Ministry of Chinese Education under Grant 20121103120020. *Asterisk indicates corresponding author.*

*H.-G Han is with the College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China (e-mail: Rechardhan@sina.com).

L.-D Wang is with the College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China (e-mail: Wanglidan01@163.com).

J.-F Qiao is with the College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China (e-mail: isibox@sina.com).

Hierarchical Extreme Learning Machine for Feedforward Neural Network

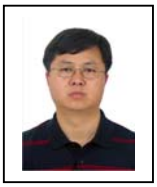
Honggui Han received the B.S. degree from Civil Aviation University of China in 2005; received M.E. and Ph.D. degrees in control engineer from Beijing University of Technology, Beijing, China, in 2007 and 2011, respectively. From 2011, he joined Beijing University of Technology where he is current an associate Professor. His current research interests include neural networks, intelligent systems, modeling and control in complex process, pattern identification, intelligent system and so on.

Communication address: College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China. E-mail: Rechardhan@sina.com. Tel: 8610-67391631

Mr. Han is a member of the IEEE Computational Intelligence Society. He is currently a reviewer of Neural Networks, Control Engineering Practice, IEEE Transactions on Neural Networks and Learning Systems, and IEEE Transactions on Fuzzy Systems.

Lidan Wang received the B.S. degree from Dezhou University in 2012; he is a M.S. student in Beijing University of Technology.

Communication address: College of Electronic and Control Engineering, Beijing University of Technology, Beijing, China. E-mail: Wanglidan01@163.com.



Jun-Fei Qiao received B.E. and M.E. degrees in control engineer from Liaoning Technical University, Fu'xin, China, in 1992 and 1995, respectively; and the Ph.D. degree from Northeast University, Shenyang, China, in 1998.

From 1998 to 2000, he was a Postdoctoral Fellow with the School of Automatics, Tianjin University, Tianjin, China. He joined Beijing University of Technology, Beijing, China, in 2000, where he is currently a Professor. He is the director of the intelligence systems lab. His research interests include neural networks, intelligent systems, self-adaptive, learning systems, and process control systems.

Communication address: College of Electronic and Control Engineering, Beijing University of Technology, Beijing, E-mail: isibox@sina.com.

Abstract—An approach, named extended extreme learning machine (ELM), is proposed for training the weights of a class of hierarchical feedforward neural network (HFNN). Unlike conventional single-hidden-layer feedforward networks (SLFNs), this hierarchical ELM (HELM) is based on the hierarchical structure which is capable of hierarchical learning of sequential information online, and one may simply choose hidden layers and then only need to adjust the output weights linking the hidden layer and the output layer. In such HELM implementations, the extended ELM provides better generalization performance during the learning process. Moreover, the proposed extended ELM method is efficient not only for HFNNs with sigmoid hidden nodes but also for HFNNs with radial basis function (RBF) hidden nodes. Finally, the HELM is applied to the activated sludge wastewater treatment processes (WWTPs) for predicting the water qualities. Experimental results and the performance comparison demonstrate the effectiveness of the proposed HELM.

Keywords—Hierarchical extreme learning machine; feedforward neural network; wastewater treatment process; predicting water qualities.