Desalination and Water Treatment
www.deswater.com

doi: 10.5004/dwt.2019.23360

140 (2019) 132–142
February

Soft sensing of effluent ammonia nitrogen using rule automatic formation-based adaptive fuzzy neural network

Hongbiao Zhou[a,b,c], Junfei Qiao[a,b,*]

[a]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China, Tel: +13801263545; emails: junfeiq123@163.com, junfeiq@bjut.edu.cn (J. Qiao); Tel. +15189544918; email: hyitzhb@163.com (H. Zhou)*
[b]*Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China*
[c]*Faculty of Automation, Huaiyin Institute of Technology, Huaian 223003, China*

Received 2 May 2018; Accepted 27 October 2018

ABSTRACT

This paper proposes a data-driven soft-sensing method for predicting effluent ammonia nitrogen (NH_4–N) in the wastewater treatment process (WWTP). In this method, a rule automatic formation-based adaptive fuzzy neural network (RAF-AFNN) is designed. The RAF algorithm, which consists of rule self-splitting strategy and fuzzy Gaussian kernel clustering, is used to automatically partition the input space and adaptively extract the most suitable fuzzy rules. An improved adaptive Levenberg–Marquardt learning algorithm is implemented to tune the parameters of the RAF-AFNN for improving prediction accuracy. An analysis of the convergence is also provided in this paper, which can guarantee the successful application of the proposed RAF-AFNN. Finally, experimental hardware, constructed from an online sensor array and via the soft-sensing method, is used to assess the effectiveness of the RAF-AFNN for solving the problem of effluent NH_4–N prediction in the WWTP. Experimental results indicate that the proposed RAF-AFNN-based soft-sensing method can predict the effluent NH_4–N precisely.

Keywords: Effluent ammonia nitrogen; Wastewater treatment process; Fuzzy neural network; Rule automatic formation; Improved adaptive LM algorithm

1. Introduction

Effluent ammonia nitrogen (NH_4–N) is a key water quality parameter used in the wastewater treatment process (WWTP); it often exceeds the standard limits because of the impact that high concentrations of influent NH_4–N have on the WWTP. Research indicates that NH_4–N pollution is one of the major factors in water eutrophication [1]. It is helpful to detect the concentration of effluent NH_4–N in a timely manner to implement the necessary control measures and prevent the deterioration of water quality [2]. At present, there are several methods that detect the effluent NH_4–N in wastewater treatment plants, for example, Nessler's reagent colorimetry, ammonia-sensitive electrode, and spectrophotometry. Although these methods have high detection precision, they need to be performed in a special laboratory [3]. Thus, these methods are cumbersome, time-consuming, and have a serious feedback delay of test results. In this case, abnormalities of effluent NH_4–N cannot be found in time, which affects the implementation of relevant remedial measures. In fact, some kinds of chemical principle-based online NH_4–N monitors are available in the market. Unfortunately, the purchasing and maintenance costs of these monitors are too high, which makes their intensive use in wastewater treatment plants uneconomical [4].

To circumvent the problems of electrochemical analysis and access to online instrumentation, the soft-sensing method is becoming increasingly prevalent in the detection of key water quality parameters in the WWTP [5,6]. With the help of data-driven soft-sensing methods, the hard-to-measure

* Corresponding author.

variables can be estimated via easy-to-measure auxiliary variables in practical industrial processes [7]. Generally, neural networks and fuzzy systems are the two most widely used soft-sensing methods [8,9]. In a study by Canete et al. [10], a three-layer multilayer perceptron (MLP) is used to design a soft-computing system that can predict the chemical oxygen demand in a WWTP. The simulation results show that the prediction accuracy of MLP is better than that of support vector machine. In a study by Nezhad et al. [11], a feed-forward backpropagation neural network is adopted to develop a prediction model for effluent quality indices in the southern Tehran municipal wastewater treatment plant. The simulation results demonstrate that the network based on Levenberg–Marquardt (LM) learning algorithm has high modeling accuracy. In a study by Bagheri et al. [12], MLP and radial basis function (RBF) neural networks are applied to design the sludge bulking pre-warning system by predicting the sludge volume index. Experimental results indicate that the MLP with genetic algorithm-based parameter optimization has lower modeling error. In a study by Dovžan et al. [13], an evolving fuzzy model is presented to monitor the dissolved oxygen (DO) concentration in a WWTP. Experimental results show that the combination of online clustering-based structure identification strategy and recursive least squares-based parameter learning algorithm can achieve good prediction accuracy. Because the initial parameters of these neural network-based prediction models are randomly generated and the node number of hidden layer is determined by trial-and-error, these networks have the disadvantages of low generalization ability. For fuzzy logical systems, the generation of suitable fuzzy rules requires extensive knowledge from domain experts.

Recent studies have shown that fuzzy neural networks (FNNs), which combine the learning capability of a neural network with the interpretability of a fuzzy system, have tremendous advantages in data-driven soft-sensing. In general, FNN-based soft-sensing methods can be classified into two categories: (1) offline soft sensing [14] and (2) online soft sensing [15,16]. In offline soft sensing, data-clustering approaches, such as *K*-means [17,18], fuzzy *C*-means (FCM) [19,20], Gustafson–Kessel [21], *K*-nearest neighbors [22], and fuzzy min–max [23], are used to generate fuzzy rules firstly. Then, local search approaches, such as least squares algorithm, error backpropagation (EBP) or their other variations, are employed to update FNN parameters. It is hypothesized that all training samples are available before using these methods. In a study by Tang et al. [24], an FNN, in which an improved least square estimation is used to optimize the linear parameters and an adaptive learning algorithm is adopted to update membership functions (MFs) and rule base, is proposed to predict lane-changing behavior. In a study by Tang et al. [25], an FNN with *K*-means clustering and weighted recursive least square estimation is proposed to predict travel speed for multistep ahead. In a study by Tang et al. [26], an enhanced evolving fuzzy neural inference system is present to estimate the travel time. In a study by Zhang and Tao [27], a novel FNN framework is proposed by combining an autoregressive with exogenous input with the nonlinear tanh function in the Takagi–Sugeno–Kang (TSK)-type fuzzy consequent part. For online soft sensing, structure learning and parameter optimization are performed immediately and simultaneously after each input–output sample pair arrives [28,29]. In a study by Wu and Er [30], a dynamic fuzzy neural network (DFNN), which functionally is equivalent to a TSK fuzzy system, is designed on the basis of extended radial basis function neural network. In DFNN, the growing of fuzzy rules relies on the system error and the accommodation boundary criterion, and the pruning of fuzzy rules depends on the significance of rules computed by the error reduction ratio algorithm. In a study by Wu et al. [31], to make up for the deficiencies in the centers and widths of the Gaussian MF in the DFNN, a generalized dynamic fuzzy neural network (GDFNN) based on ellipsoidal basis function is presented. In a study by Ma et al. [32], the GDFNN proposed by Wu et al. [31] is adopted to forecast the short-term wind speed. The main challenge of the online modeling methods is to design the growing and pruning mechanisms to obtain fuzzy rules.

This study mainly focuses on the design of a soft-sensing model for effluent NH_4–N using FCM clustering method and FNN. However, the generic FCM algorithm is difficult to achieve satisfactory performance in the linearly inseparable input space. Therefore, setting the number of fuzzy rules in advance becomes necessary, which decreases the clustering performance. Additionally, the EBP learning algorithm widely applied in the FNNs as parameter optimizers has some shortcomings, including slow convergence speed and falling into a local optimum. To circumvent some of the problems discussed earlier in this paper, a rule automatic formation-based adaptive fuzzy neural network (RAF-AFNN) method is presented to develop the prediction model of the effluent NH_4–N in a WWTP. First, an RAF algorithm, based on the fuzzy Gaussian kernel clustering (FGKC), is used to automatically partition the input space and further extract the initial fuzzy rules. Second, to speed up the convergence in the training process, an improved adaptive Levenberg–Marquardt (IALM) learning algorithm is adopted to adjust the network parameters. Finally, an experimental system, which includes an online sensor array with the proposed RAF-AFNN soft-sensing approach, is developed to predict effluent NH_4–N in a real-time WWTP. Experimental results show that the RAF-AFNN-based soft-sensing method can achieve superior performance for effluent NH_4–N prediction.

2. Data description

In this section, a soft-sensing system, which consists of an online sensor array and a prediction model, is developed to assess the effectiveness of the RAF-AFNN method. The schematic diagram of the soft-sensing system for predicting effluent NH_4–N in the WWTP is shown in Fig. 1.

In the prediction system, the input variables are the variables that can be measured easily, while the output variable is the effluent NH_4–N. In this study, the RAF-AFNN method is applied to establish a nonlinear mapping relationship between input–output by using data samples. Table 1 lists the WWTP process variables. To reduce the redundancy caused by excessive input variables, the partial least squares algorithm is applied to complete the feature variable selection step in the data preprocessing phase. According to the result of variable selection, total suspended solids (TSS) and effluent total phosphorus (TP) are redundant variables, water temperature (*T*), DO

concentration, effluent pH, effluent oxidation–reduction potential (ORP), and effluent nitrate nitrogen (NO_3–N) concentration are selected as the input variables. In this study, 1,000 input–output data samples, measured between 1 June 2015 and 31 July 2015, were collected from a real small-scale wastewater treatment plant in Beijing, China. In order to validate the prediction performance of different algorithms, data samples are divided into two parts: training data set and testing data set. After performing wavelet package de-noising and normalization, every other nine data points, one data samples is chosen from the entire data set to form the testing data set (a total of 100 data samples), and the remaining 900 data samples are applied as the training data set. During the testing process, the input values of the testing samples are shown in Fig. 2. As can be seen from Fig. 2, the effluent NH_4–N has time-varying and nonlinear characteristics because of the complex biochemical reactions in the WWTP and the strong interaction between the input variables. Obviously, depending on mechanism model or linear data analysis methods, it is impossible to accurately predict the values of effluent NH_4–N. In order to improve the prediction accuracy, this paper uses AFNN with RAF strategy to establish reliable soft-sensing model of effluent NH_4–N.

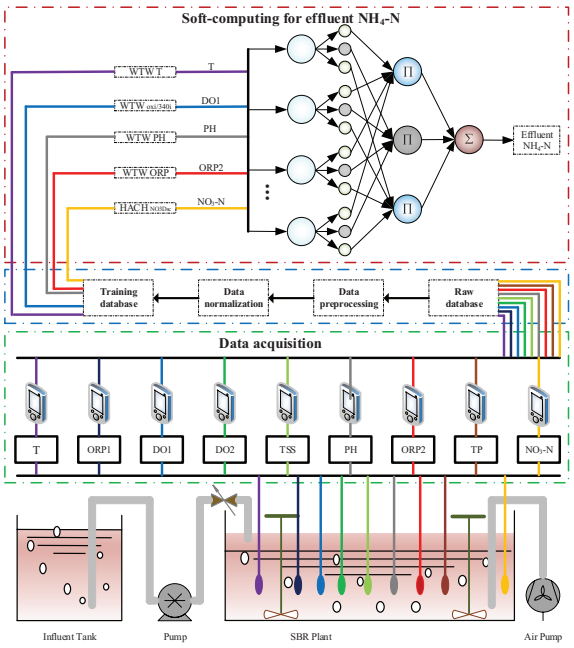

Fig. 1. The soft-sensing system of effluent NH_4–N.

Table 1
Variables involved in the WWTP

Variables	Main apparatus and instruments	Locations
T, °C	WTW MIQ/TC 2020XT T probe	Anaerobic tank
ORP1, mV	WTW MIQ/TC 2020XT ORP probe	Anaerobic tank
DO1, mg/L	WTW oxi/340i oxygen probe	Aerobic front
DO2, mg/L	WTW oxi/340i oxygen probe	Aerobic end
TSS, g/L	7110 MTF-FG SS/TSS	Aerobic end
pH	WTW inoLab pH level2	Effluent outlet
ORP2, mV	WTW MIQ/TC 2020XT ORP probe	Effluent outlet
TP, mg/L	HACH NPW160	Effluent outlet
NO_3–N, mg/L	HACH NO3Dsc	Effluent outlet
NH_4–N, mg/L	HACH Amtax Compact II	Effluent outlet

3. Methodology

3.1. Architecture of the FNN

In this section, we firstly introduce the architecture of the FNN. It can be seen from Fig. 3, the FNN has four layers, that is, the input layer, the MF layer, the rule layer, and the output layer [31]. The following is the mathematical description of this FNN:

1. *The input layer:* In this layer, there are *n* nodes, and each of them represents an input linguistic variable. The output values of input layer are as follows:

$$u_i = x_i, \quad i = 1,2,...,n \tag{1}$$

where u_i is the output value of the *i*th node, and $x = [x_1, x_2, \ldots, x_n]^T$ is the input vector.

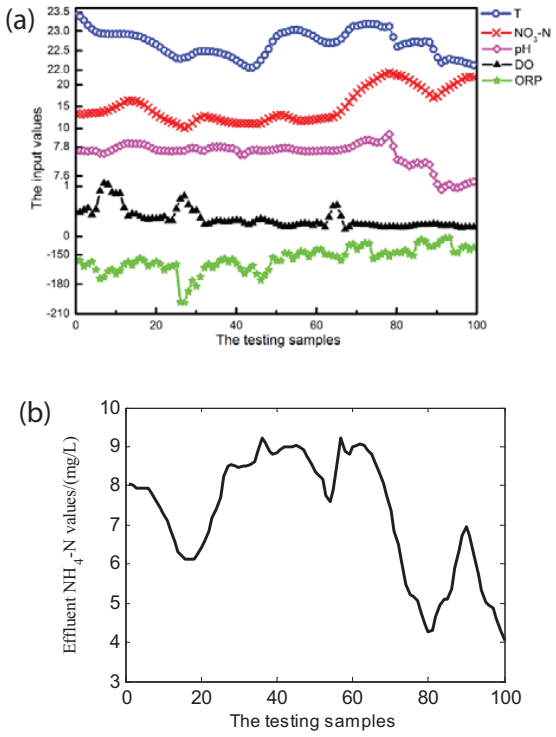

Fig. 2. The (a) input and (b) output values of the testing samples.

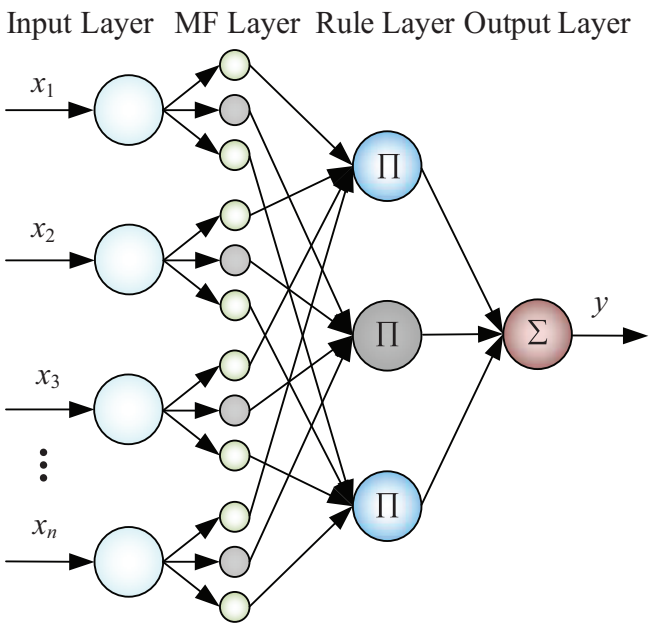

Fig. 3. Architecture of the FNN.

2. *The MF layer:* In this layer, there are $n \times r$ nodes, and each of them represents an MF which is in the form of Gaussian function. The output values of MF layer can be expressed as follows:

$$\mu_{ij}(x_i) = \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right], \quad i = 1,2,...,n; \quad j = 1,2,...,r \tag{2}$$

where μ_{ij} is the value of the jth MF of x_i, c_{ij}, and σ_{ij} denotes the center and width of the jth MF of x_i, respectively.

3. *The rule layer:* In this layer, there are r nodes, and each of them represents a premise of a fuzzy rule. The output of the jth rule neuron can be computed as follows:

$$\varphi_j(x) = \exp\left[-\sum_{i=1}^{n}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right], \quad j = 1,2,...,r \tag{3}$$

And the normalized output is as follows:

$$h_j(x) = \frac{\varphi_j(x)}{\sum_{k=1}^{r}\varphi_k(x)} = \frac{\exp\left[-\sum_{i=1}^{n}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right]}{\sum_{k=1}^{r}\exp\left[-\sum_{i=1}^{n}\frac{(x_i - c_{ik})^2}{\sigma_{ik}^2}\right]}, \quad j = 1,2,...,r \tag{4}$$

where $h = [h_1, h_2, \ldots, h_r]$ is the normalized output vector of the rule layer.

4. *The output layer:* For a multi-input single-output system, there is a single-output node in this layer. The output value is the weighted summation of incoming signals and is given by the following equation:

$$y(x) = \sum_{j=1}^{r} w_j h_j = \frac{\sum_{j=1}^{r} w_j \exp\left[-\sum_{i=1}^{n}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right]}{\sum_{k=1}^{r}\exp\left[-\sum_{i=1}^{n}\frac{(x_i - c_{ik})^2}{\sigma_{ik}^2}\right]} \tag{5}$$

where w_j is the consequent of the jth rule.

3.2. Rule automatic formation-based adaptive fuzzy neural network

Two important issues associated with the construction of an FNN are the structure identification and the parameter estimation. For FNN, a good parameter initialization algorithm can accelerate the convergence speed and increase the chance of getting out of a local optimum. In this section, the initial fuzzy rules are adaptively constructed by an RAF algorithm, while the parameters are adjusted by an IALM algorithm.

3.2.1. Structure learning of the RAF-AFNN

When extracting fuzzy rules by using clustering algorithms, a cluster corresponds to a fuzzy rule. To obtain the most suitable fuzzy rules adaptively, this paper proposes an RAF algorithm, which is composed of a rule self-splitting strategy and a FGKC algorithm.

3.2.1.1. RAF strategy In the RAF strategy, the mean variance of a cluster is applied to as the criterion of rule self-splitting [18]. Suppose that there are r clusters at present, the one with the largest mean variance must be found firstly.

$$I = \arg \max_{1 \le j \le r} \bar{s}_j^2 \tag{6}$$

where $\bar{s}_j^2$ denotes the mean variance of the jth cluster, which can be calculated by the sum of variance of each input variable as follows:

$$\bar{s}_j^2 = \sum_{i=1}^{n} s_{ij}^2 \tag{7}$$

where s_{ij} is the variance of the ith dimension of the jth cluster, which is given as follows:

$$s_{ij}^2 = \frac{1}{P_j}\sum_{x \in v_j}(x_{ij} - v_{ij})^2 \tag{8}$$

where $x \in v_j$ indicates the sample x that takes from the jth cluster, v_{ij} is the ith dimension of the center of the jth cluster, and P_j is the number of samples involved in the jth cluster.

If $\bar{s}_I^2 > s_{th}$ (s_{th} is a user-defined threshold), the Ith cluster will be divided into two new clusters and $r = r + 1$. Suppose that the center of the Ith cluster is $\bar{v}_I = (v_{I1}, v_{I2}, ..., v_{In})$ and the centers of the two newly formed clusters are $\bar{v}_I + \bar{\alpha}$ and $\bar{v}_I - \bar{\alpha}$, respectively, and where $\bar{\alpha}$ is a small value. In this paper, $\bar{\alpha}$ is set to be 1% of the range [0,1], that is, $\bar{\alpha} = 0.01$. After the rule-splitting operation is performed, the new centers of the r clusters are iteratively calculated by the FGKC algorithm. The entire process is repeated until $\bar{s}_I^2 < s_{th}$. Noted that the center of the first cluster is the average of all training samples.

3.2.1.2. FGKC algorithm FCM which is an enhanced version of *K*-means is the most widely used algorithm in the clustering field [33]. When the boundary of the sample cluster is difficult to determine, FCM is apt to fall into a local minimum [34]. In this study, the FGKC algorithm [35,36], where the Euclidean distance adopted in FCM is replaced by a Gaussian kernel function, is used to extract the most suitable fuzzy rules from effluent NH_4–H training data set.

FGKC can obtain membership value of each sample to each cluster center by minimizing a quadratic objective function, to achieve automatic classification of data set. Let $X = \{x_1, x_2, \ldots, x_P\}$ be a training data set and $V = \{v_1, v_2, \ldots, v_C\}$ be the cluster centers. The optimization model of FGKC is defined as follows:

$$J_m = \sum_{c=1}^{C}\sum_{p=1}^{P}\mu_{pc}^m \left\|\Phi(x_p) - \Phi(v_c)\right\|^2$$
$$\text{s.t.} \quad \sum_{c=1}^{C}\mu_{pc} = 1, \quad 0 \le \mu_{pc} \le 1 \tag{9}$$

where J_m is the objective function, μ_{pc} is the membership value of the cth cluster center for the pth data sample, P is the number of data samples, C is the number of cluster centers, $m > 1$ is the fuzziness index (in general, $m = 2$), and Φ is the feature mapping function. According to the transformation rules of kernel method, we obtain the following equation:

$$\begin{aligned}\left\|\Phi(x_p) - \Phi(v_c)\right\|^2 &= (\Phi(x_p) - \Phi(v_c))^T(\Phi(x_p) - \Phi(v_c)) \\ &= \Phi(x_p)^T\Phi(x_p) - \Phi(x_p)^T\Phi(v_c) - \Phi(v_c)^T\Phi(x_p) + \Phi(v_c)^T\Phi(v_c) \\ &= K(x_p, x_p) + K(v_c, v_c) - 2K(x_p, v_c)\end{aligned} \tag{10}$$

where the Gaussian kernel function K is defined as follows:

$$K(x, v) = \exp\left[-\frac{\|x - v\|^2}{b^2}\right] \tag{11}$$

where b is the kernel bandwidth. According to $K(x, x) = 1$, the objective function (9) can be rewritten as follows:

$$J_m = 2\sum_{c=1}^{C}\sum_{p=1}^{P}\mu_{pc}^m(1 - K(x_p, v_c)) \tag{12}$$

Then, optimal solutions for the objective function (12) can be obtained by updating the clustering centers and membership matrix as follows:

$$v_c = \frac{\sum_{p=1}^{P}\mu_{pc}^m K(x_p, v_c)x_p}{\sum_{p=1}^{P}\mu_{pc}^m K(x_p, v_c)} \tag{13}$$

and

$$\mu_{pc} = \frac{(1 - K(x_p, v_c))^{-1/(m-1)}}{\sum_{c=1}^{C}(1 - K(x_p, v_c))^{-1/(m-1)}} \tag{14}$$

3.2.1.3. Procedure of the RAF strategy The pseudo-code of the proposed RAF algorithm is described as follows:

```
Algorithm: RAF
Set cluster number r = 1, calculate the center of the first cluster v̄_1;
Calculate the mean variance s̄_j^2 by Eqs. (7) and (8);
while (1)
  if s̄_I^2 < s_th then
    break;
  else
    Split the Ith cluster and obtain two new centers;
    r = r + 1;
  end
  while (1)
    if current iteration times t_1 > T_1 or |J_m(t_1) − J_m(t_1−1)| < ε then
      break;
    else
      Assign each input training data to cluster v_c by Eq. (9);
      Recompute the center of cluster v_c by Eq. (13);
      t_1++;
    end
  end
end
return r clusters;
```

3.2.2. Parameter learning of the RAF-AFNN

In this paper, to enhance the convergence rate and generalization performance, an IALM algorithm is developed to train the network parameters (i.e., centers, widths, and weights). According to the conventional LM algorithm [37], the update rule is presented as follows:

$$\Theta(t+1) = \Theta(t) - (J(t)^T J(t) + \eta(t)I)^{-1} J(t)^T e(t) \tag{15}$$

where $\Theta(t) = [w(t), c(t), \sigma(t)]$ is the parameter vector, J is the Jacobian matrix, $\eta(t)$ is the learning coefficient, I is the

identity matrix that is adopted to circumvent the difficulty of singularity in reversing the matrix, and $e(t) = [e_1(t), e_2(t), \ldots, e_p(t)]^T$ is the error vector. For the pth sample, the error between the desired output and network output of FNN is defined as follows:

$$e^p(t) = y_d^p(t) - y^p(t), \quad p = 1, 2, \ldots, P \tag{16}$$

where P is the total number of the samples.

The Jacobian matrix is given as follows:

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1}, \ldots, \frac{\partial e_1}{\partial w_r}, \frac{\partial e_1}{\partial c_{11}}, \ldots, \frac{\partial e_1}{\partial c_{nr}}, \frac{\partial e_1}{\partial \sigma_{11}}, \ldots, \frac{\partial e_1}{\partial \sigma_{nr}} \\ \frac{\partial e_2}{\partial w_1}, \ldots, \frac{\partial e_2}{\partial w_r}, \frac{\partial e_2}{\partial c_{11}}, \ldots, \frac{\partial e_2}{\partial c_{nr}}, \frac{\partial e_2}{\partial \sigma_{11}}, \ldots, \frac{\partial e_2}{\partial \sigma_{nr}} \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ \frac{\partial e_p}{\partial w_1}, \ldots, \frac{\partial e_p}{\partial w_r}, \frac{\partial e_p}{\partial c_{11}}, \ldots, \frac{\partial e_p}{\partial c_{nr}}, \frac{\partial e_p}{\partial \sigma_{11}}, \ldots, \frac{\partial e_p}{\partial \sigma_{nr}} \end{bmatrix} \tag{17}$$

From Eq. (17), it can be seen that the number of rows of Jacobian matrix is equal to the number of training samples, and the number of columns of Jacobian matrix is equal to the number of parameters. Hence, the computational burden and storage capacity will increase dramatically as the number of training samples increases.

To circumvent the problem, an IALM algorithm is presented to optimize all the parameters in the FNN [37,38]. The parameter vector $\Theta(t)$ is updated as follows:

$$\Theta(t+1) = \Theta(t) - (\Psi(t) + \eta(t)\mathrm{I})^{-1}\Omega(t) \tag{18}$$

where $\Psi(t)$ is the quasi-Hessian matrix, and $\Omega(t)$ is the gradient vector. The adaptive learning rate $\eta(t)$ is given as follows:

$$\eta(t) = \beta\|e(t)\| + (1-\beta)\|\Omega(t)\| \tag{19}$$

where $\beta(0 < \beta < 1)$ is a user-defined constant. Furthermore, the $\Psi(t)$ and $\Omega(t)$ are the accumulation of submatrices and subvectors for all samples, respectively.

$$\Psi(t) = \sum_{p=1}^{P} \Psi_p(t) \tag{20}$$

$$\Omega(t) = \sum_{p=1}^{P} \omega_p(t) \tag{21}$$

where the submatrices $\psi_p(t)$ and the subvectors $\omega_p(t)$ are defined as follows:

$$\Psi_p(t) = j_p^T(t) j_p(t) \tag{22}$$

$$\omega_p(t) = j_p^T(t) e_p(t) \tag{23}$$

where $j_p(t)$ is the row vector of Jacobian matrix.

$$j_p(t) = \begin{bmatrix} \frac{\partial e_p(t)}{\partial w_1(t)}, \ldots, \frac{\partial e_p(t)}{\partial w_r(t)}, \frac{\partial e_p(t)}{\partial c_{11}(t)}, \ldots, \frac{\partial e_p(t)}{\partial c_{ij}(t)}, \ldots, \\ \frac{\partial e_p(t)}{\partial c_{nr}(t)}, \frac{\partial e_p(t)}{\partial \sigma_{11}(t)}, \ldots, \frac{\partial e_p(t)}{\partial \sigma_{ij}(t)}, \ldots, \frac{\partial e_p(t)}{\partial \sigma_{nr}(t)} \end{bmatrix} \tag{24}$$

According to the update rule of gradient descent learning approach, the elements of the Jacobian row vector are presented as follows:

$$\frac{\partial e_p(t)}{\partial w_j(t)} = \frac{\partial e_p(t)}{\partial y_p(t)} \frac{\partial y_p(t)}{\partial w_j(t)} = -h_j(t) \tag{25}$$

$$\frac{\partial e_p(t)}{\partial c_{ij}(t)} = \frac{\partial e_p(t)}{\partial y_p(t)} \frac{\partial y_p(t)}{\partial h_j(t)} \frac{\partial h_j(t)}{\partial v_j(t)} \frac{\partial v_j(t)}{\partial \mu_{ij}(t)} \frac{\partial \mu_{ij}(t)}{\partial c_{ij}(t)} = -w_j(t) \frac{\sum_{i \neq j}^{r} v_i(t)}{(\sum_{i=1}^{r} v_i(t))^2} \prod_{k \neq i} \mu_{kj}(t) \frac{\partial \mu_{ij}(t)}{\partial c_{ij}(t)} \tag{26}$$

$$\frac{\partial e_p(t)}{\partial \sigma_{ij}(t)} = \frac{\partial e_p(t)}{\partial y_p(t)} \frac{\partial y_p(t)}{\partial h_j(t)} \frac{\partial h_j(t)}{\partial v_j(t)} \frac{\partial v_j(t)}{\partial \mu_{ij}(t)} \frac{\partial \mu_{ij}(t)}{\partial \sigma_{ij}(t)} = -w_j(t) \frac{\sum_{i \neq j}^{r} v_i(t)}{(\sum_{i=1}^{r} v_i(t))^2} \prod_{k \neq i} \mu_{kj}(t) \frac{\partial \mu_{ij}(t)}{\partial \sigma_{ij}(t)} \tag{27}$$

where

$$\frac{\partial \mu_{ij}(t)}{\partial c_{ij}(t)} = \frac{2(x_i(t) - c_{ij}(t)) \exp(-(x_i(t) - c_{ij}(t))^2 / \sigma_{ij}^2(t))}{\sigma_{ij}^2(t)},$$

$$\frac{\partial \mu_{ij}(t)}{\partial \sigma_{ij}(t)} = \frac{2(x_i(t) - c_{ij}(t))^2 \exp(-(x_i(t) - c_{ij}(t))^2 / \sigma_{ij}^2(t))}{\sigma_{ij}^3(t)}$$

It should be noted that for the computation of submatrices $\psi_p(t)$ and subvectors $\omega_p(t)$, only $(2n + 1) \times r$ elements of $j_p(t)$ need to be calculated for each pattern p separately. In the IALM algorithm, there is no need to store and multiply the Jacobian matrix, both quasi-Hessian matrix $\Psi(t)$ and gradient vector $\Omega(t)$ can be calculated directly, resulting in the reduction of memory cost and computation complexity. Meanwhile, during the learning process, the adaptive learning rate strategy defined in Eq. (19) can help to accelerate the learning process and enhance the generalization capability.

3.2.3. Procedure of the RAF-AFNN

The pseudo-code of the proposed RAF-AFNN method is summarized as follows:

Algorithm: RAF-AFNN	
1	Initialization, set the maximum iteration times T_1 and T_2;
2	Obtain r clusters using RAF algorithm;

(*Continued*)

Algorithm: RAF-AFNN (*Continued*)

```
Create an initial four-layer FNN. The initial centers and
widths of Gaussian functions are the corresponding
cluster centers and their variances. The initial output
weights are obtained using least square method.
while (1)
  if current iteration times t2 > T2 then
    break;
  else
    for p = 1:P do
      Calculate the outputs of FNN yp(t2) by Eq. (5);
      Make the error of each sample ep(t2) by Eq. (16);
      Make the submatrices ψp(t) by Eq. (22);
      Make the subvectors ωp(t) by Eq. (23);
    end
    Update the adaptive learning rate η(t) by Eq. (19);
    Make the quasi-Hessian matrix Ψ(t2) by Eq. (20);
    Make the gradient vector Ω(t2) by Eq. (21);
    Update the parameter vector Θ(t2) by Eq. (18).
    t2++;
  end
end
```

4. Convergence analysis

For a learning system based on neural network, convergence ability is one of the preconditions for the successful applications. The convergence of RAF-AFNN is mainly influenced by the initialization parameters, the learning rate, and the characteristics of quasi-Hessian matrix. Because a set of parameters closed to the optimal parameters can be found by the RAF-based parameter initialization method and the nonsingular and positive definite of the quasi-Hessian matrix $\Psi(t)$ can be guaranteed by the adaptive learning rate defined in Eq. (19), the convergence of RAF-AFNN can be maintained (i.e., $e(t) \to 0$ as $t \to +\infty$) when the parameters of RAF-AFNN are updated by Eq. (18), if

$$\left\|\Delta\Theta(t)\right\| \le \min\left\{\left\|\Delta\Theta(t-1)\right\|, \frac{\left\|\Omega(\Theta(t-1))\right\|}{\left\|\Psi(\Theta(t-1))\right\|}\right\} \tag{28}$$

Proof: For the IALM algorithm defined as follows:

$$\begin{aligned}\Theta(t+1) &= \Theta(t) - (\Psi(t) + \eta(t)\mathrm{I})^{-1}\Omega(t) \\ &= \Theta(t) - \mathrm{H}(t)^{-1}\Omega(t)\end{aligned} \tag{29}$$

Assume that Θ^* is the optimal parameter vector and $\Omega(Q^*) = 0$, then we have the following:

$$\begin{aligned}\left\|\Theta(t+1) - \Theta^*\right\| &= \left\|\Theta(t) - \mathrm{H}(t)^{-1}\Omega(t) - \Theta^*\right\| \\ &= \left\|\Theta(t) - \Theta^* - \mathrm{H}(t)^{-1}\left[\Omega(t) - \Omega(\Theta^*)\right]\right\| \\ &= \left\|\mathrm{H}(t)^{-1}\left[\Omega(t) - \Omega(\Theta^*) - \mathrm{H}(t)\left(\Theta(t) - \Theta^*\right)\right]\right\|\end{aligned} \tag{30}$$

According to the matrix inequality, the Eq. (30) can be rewritten as follows:

$$\left\|\Theta(t+1) - \Theta^*\right\| \le \left\|\mathrm{H}(t)^{-1}\right\|\left\|\Omega(t) - \Omega(\Theta^*) - \mathrm{H}(t)\left(\Theta(t) - \Theta^*\right)\right\| \tag{31}$$

Assume that $\Omega(t)$ is continuously differentiable and $H(t)$ is Lipschitz continuous on some neighborhood of Θ^*, then we have the following:

$$\left\|\Theta(t+1) - \Theta^*\right\| \le \frac{\beta}{\lambda_{\min}^t}\left\|\Theta(t) - \Theta^*\right\|^2 \tag{32}$$

where β is a positive constant and $\lambda^t_{\min}$ is the smallest eigenvalue of $H(t)$. Similarly, we can derive the following formulas:

$$\left\|\Theta(t) - \Theta^*\right\| \le \frac{\beta}{\lambda_{\min}^{t-1}}\left\|\Theta(t-1) - \Theta^*\right\|^2 \tag{33}$$

$$\left\|\Theta(2) - \Theta^*\right\| \le \frac{\beta}{\lambda_{\min}^{1}}\left\|\Theta(1) - \Theta^*\right\|^2 \tag{34}$$

Thus, we obtain the following:

$$\left\|\Theta(t+1) - \Theta^*\right\| \le \frac{\beta^{2t+1}}{\left(\lambda_{\min}^1\lambda_{\min}^2\ldots\lambda_{\min}^{t-1}\right)^2\lambda_{\min}^t}\left\|\Theta(t) - \Theta^*\right\|^{2t} \tag{35}$$

Based on Eq. (35), the IALM algorithm can finally converge to the optimal parameter vector Θ^*. Below, we use Lyapunov stability theorem to prove the convergence of the proposed RAF-AFNN. Suppose that the Lyapunov function is defined as follows:

$$V(\Theta(t)) = \frac{1}{2}e^T(t)e(t) \tag{36}$$

where $e(t) = [e_1(t), e_2(t), \ldots, e_p(t)]^T$ and $V(\Theta(t)) > 0$. In terms of the studies by Wilamowski and Yu [37] and Han et al. [38], the deviation of Lyapunov function is given as follows:

$$\begin{aligned}\Delta V(\Theta(t)) &= V(\Theta(t+1)) - V(\Theta(t)) \\ &= -\nabla\mathrm{E}^T(\Theta(t))\Delta\Theta(t) + \frac{1}{2}\Delta\Theta^T(t)\nabla^2\mathrm{E}(\Theta(t))\Delta\Theta(t)\end{aligned} \tag{37}$$

where $\nabla E(\Theta(t))$ and $\nabla^2 E(\Theta(t))$ represent the gradient vector and Hessian matrix of the objective function, respectively. According to the update rule given in Eq. (16), the following is obtained:

$$\begin{cases}\Delta\Theta(t) = (\Psi(\Theta(t)) + \eta(t)\mathrm{I})^{-1}\Omega(\Theta(t)), \\ \nabla\mathrm{E}(\Theta(t)) = \Omega(\Theta(t)), \\ \nabla^2\mathrm{E}(\Theta(t)) = \Psi(\Theta(t)) + \eta(t)\mathrm{I}\end{cases} \tag{38}$$

From Eqs. (37) and (38), $\Delta V(\Theta(t))$ can be expressed as follows:

$$\Delta V(\Theta(t)) = -\frac{1}{2}\Delta\Theta^T(t)\nabla^2 E(\Theta(t))\Delta\Theta(t) \tag{39}$$

when the condition of Eq. (28) is satisfied, we can safely draw a conclusion that $\Delta V(\Theta(t)) < 0$, because the matrix $\nabla^2 E(\Theta(t))$ is positive definite. Therefore, the following is obtained:

$$\lim_{t\to+\infty} e(t)=0 \tag{40}$$

Hence, from the Lyapunov stability theorem, the RAF-AFNN is theoretically convergent.

5. Simulation studies

5.1. Parameters settings and evaluation indicators

By the practical test, the parameters of RAF-AFNN are set as follows: $V_{th} = 0.03$, $b = 5$, $\varepsilon = 1 \times 10^{-5}$, $T_1 = 200$, $\beta = 0.3$, and $T_2 = 1{,}000$. In this experiment, 12 initial fuzzy rules were obtained by using RAF-based rule generation method. To validate the performance of the proposed RAF-AFNN method, the root-mean-square error (RMSE) and accuracy in Eqs. (41) and (42) are used as the performance criterion. All results are averaged on 30 independent runs.

$$\text{RMSE} = \sqrt{\frac{1}{P}\sum_{p=1}^{P}(y_d^p - y^p)^2} \tag{41}$$

$$\text{Accuracy} = \frac{1}{P}\sum_{p=1}^{P}\left(1 - \frac{\left|y_d^p - y^p\right|}{\left|y_d^p\right|}\right)\times 100\% \tag{42}$$

5.2. Experimental results

The effectiveness of the RAF-AFNN is verified by comparing with other works. Note that the wavelet packet denoising is performed before inputting the data samples into the proposed learning system. Moreover, to clearly illustrate the experimental results, two cases of this experiment are discussed respectively.

5.2.1. Case 1

In this case, to demonstrate the advantages of the RAF and IALM strategies, three models designed by different combinations of each strategy are investigated, that is, RAF + IALM, FCM + IALM, and RAF + EBP. In this study, each model has the same number of fuzzy rules.

Fig. 4 shows the variation of RMSE values in RAF-AFNN during the training process. From Fig. 4, the RMSE can converge to a steady state with a low initial value. Furthermore, the predicting results obtained by the RAF-AFNN method are shown in Fig. 5, and the corresponding predicting errors are depicted in Fig. 6. Fig. 5 indicates that the proposed RAF-AFNN-based soft-sensing method can approximate the actual values of effluent NH_4–N well. Fig. 6 shows that the prediction errors are mainly within the closed interval [–0.2, 0.2], which indicates that the proposed RAF-AFNN-based soft-sensing method meets the detection requirements of effluent NH_4–N in the WWTP.

The experimental results in terms of the mean training RMSE, testing RMSE and accuracy for the abovementioned three models, the traditional FNN and two dynamic FNNs (DFNN [30], GDFNN [31]) are summarized in Table 2 for comparison.

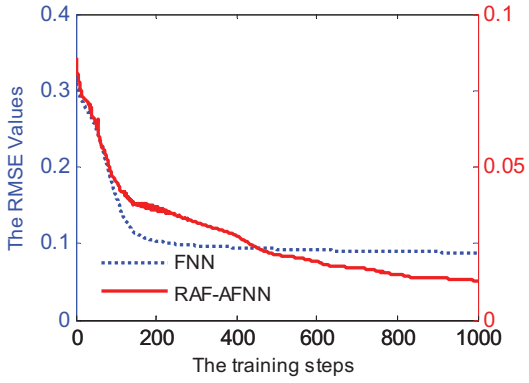

Fig. 4. RMSE during training.

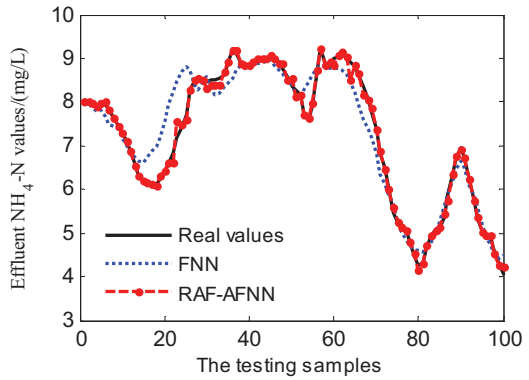

Fig. 5. The predicting results for Case 1.

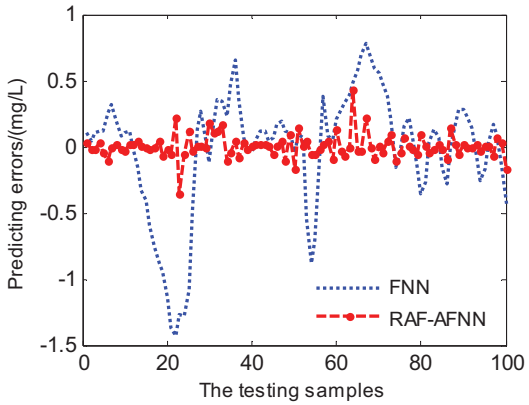

Fig. 6. The predicting errors for Case 1.

By analyzing Table 2, the advantages of the soft-sensing method, based on both RAF and IALM, are summarized as follows. First, the most suitable fuzzy rules can be automatically extracted from the input data sets by the proposed RAF algorithm and the fuzzy Gaussian kernel function. Thus, the RAF-based models have better performance than FCM-based models. Second, the proposed IALM parameter optimization algorithm can accelerate the training speed and enhance its generalization capability. The over-fitting problem, which often occurs in the neural networks, can be well controlled by the IALM-based learning algorithm. Thus, to predict the effluent NH_4–N, the soft-sensing method, designed by the combination of RAF and IALM, obtains the best mean testing RMSE and the best mean prediction accuracy.

5.2.2. Case 2

In this case, the prediction performance of the RAF-AFNN-based soft-sensing method is compared with some other models, such as the mathematical method [39], the dynamic autoregressive exogenous (ARX) method [40], the multiple linear regression (MLR) [41], the MLP [11], and the RBF [12]. The parameters of these methods are the same as the initial papers.

Fig. 7 shows the predicting results obtained by the MLR, the RBF, and the RAF-AFNN. The corresponding predicting errors are depicted in Fig. 8. From Figs. 7 and 8, the proposed RAF-AFNN-based soft-sensing method is more competitive in solving effluent NH_4–N prediction problem compared with MLR and RBF.

Table 2
Performance comparison of different methods for Case 1

Methods	Hidden nodes	Mean training RMSE	Mean testing RMSE	Mean accuracy (%)
DFNN [30]	18	0.2445	0.2536	94.74
GDFNN [31]	16	0.2488	0.2501	95.22
FNN	12	0.4718	0.4798	93.15
RAF + EBP	12	0.2862	0.2920	94.18
FCM + IALM	12	0.0951	0.0996	97.29
RAF + IALM	12	0.0684	0.0706	98.46

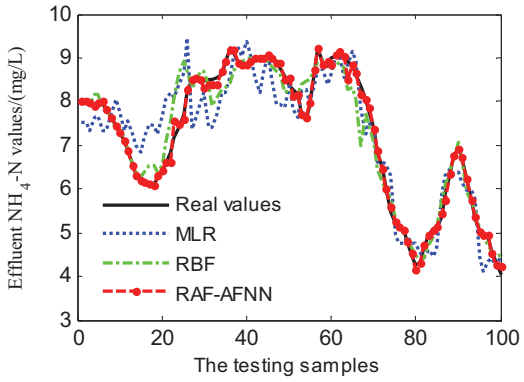

Fig. 7. The predicting results for Case 2.

To show the prediction abilities of the proposed RAF-AFNN model, the detailed comparison with other methods is summarized in Table 3. In this table, the mean testing RMSE and the mean accuracy are listed. As can be seen from Table 3, the proposed RAF-AFNN-based soft-sensing method has the smallest mean testing RMSE value. Moreover, compared with the mathematical method, the dynamic ARX method, the MLR method, the MLP method, and the RBF method, the proposed RAF-AFNN-based soft-sensing method has the highest mean prediction accuracy. The simulation results indicate that the proposed soft-sensing approach is suitable for the effluent NH_4–N modeling.

6. Conclusions

The detection of effluent NH_4–N in the WWTP depends mainly on the online instruments or offline biochemical analysis. The cost of purchasing and maintenance of these facilities is high. In this paper, to obtain the values of effluent NH_4–N in real time, an RAF-AFNN-based soft-sensing model is designed. To improve the modeling performance of the proposed RAF-AFNN approach, the RAF-based algorithm is used to extract the initial fuzzy rules. Then, the IALM-based learning algorithm is applied to adjust network parameters during the training process. In addition, experimental hardware is constructed using an online sensor

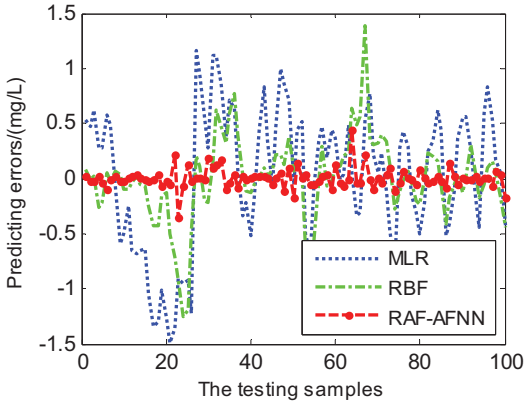

Fig. 8. The predicting errors for Case 2.

Table 3
Performance comparison of different methods for Case 2

Methods	Hidden nodes	Mean training RMSE	Mean testing RMSE	Mean accuracy (%)
Mathematic model [39]	–	–	–	90.87
Dynamic ARX [40]	–	–	–	85.12
MLR [41]	–	0.6354	0.6244	90.51
MLP [11]	12	0.4956	0.5013	92.23
RBF [12]	12	0.4284	0.4323	92.95
RAF-AFNN	12	0.0684	0.0706	98.46

array and the soft-sensing method to predict the effluent NH_4–N. The experimental results show that the RAF-AFNN can achieve superior prediction accuracy of effluent NH_4–N when compared with other prediction models.

Furthermore, the experimental results also demonstrate that the proposed RAF-AFNN-based soft-sensing method can obtain satisfactory prediction results using T, DO, ORP, NO_3–N, and pH as the input variables. According to the accurate predicting results of effluent NH_4–N, it is helpful to find the change of water quality and take the corresponding restrain measures in time. The success of RAF-AFNN-based soft-sensing method suggests that it can be used to design an effective predictive controller in the WWTP.

Acknowledgment

This work was supported in part by the National Science Foundation for Distinguished Young Scholars of China (grant no. 61225016) and the State Key Program of National Natural Science of China (grant no. 61533002).

References

[1] H.X. Du, F.S. Li, Z.J. Yu, C.H. Feng, W.H. Li, Nitrification and denitrification in two-chamber microbial fuel cells for treatment of wastewater containing high concentrations of ammonia nitrogen, Environ. Technol., 37 (2016) 1232–1239.
[2] S.F. Kosari, B. Rezania, K.V. Lo, D.S. Mavinic, Operational strategy for nitrogen removal from centrate in a two-stage partial nitrification – anammox process, Environ. Technol., 35 (2014) 1110–1120.
[3] Y.Q. Yao, D.F. Lu, Z.M. Qi, S.H. Xia, Miniaturized optical system for detection of ammonia nitrogen in water based on gas-phase colorimetry, Anal. Lett., 45 (2012) 2176–2184.
[4] I.M. Valente, H.M. Oliveira, C.D. Vaz, R.M. Ramos, Determination of ammonia nitrogen in solid and liquid high-complex matrices using one-step gas-diffusion microextraction and fluorimetric detection, Talanta, 167 (2017) 747–753.
[5] H. Haimi, M. Mulas, F. Corona, R. Vahala, Data-derived soft-sensors for biological wastewater treatment plants: an overview, Environ. Modell. Software, 47 (2013) 88–107.
[6] H.G. Han, Y. Li, Y.N. Guo, J.F. Qiao, A soft computing method to predict sludge volume index based on a recurrent self-organizing neural network, Appl. Soft Comput., 38 (2016) 477–486.
[7] X.F. Yuan, Z.Q. Ge, Z.H. Song, Y.L. Wang, C.H. Yang, H.W. Zhang, Soft sensor modeling of nonlinear industrial processes based on weighted probabilistic projection regression, IEEE Trans. Instrum. Meas., 66 (2017) 837–845.
[8] S. Yin, X.W. Li, H.J. Gao, O. Kaynak, Data-based techniques focused on modern industry: an overview, IEEE Trans. Ind. Electron., 62 (2015) 657–667.
[9] J.B. Zhang, Z.H. Deng, K.S. Choi, S.T. Wang, Data-driven elastic fuzzy logic system modeling: constructing a concise system with human-like inference mechanism, IEEE Trans. Fuzzy Syst., 26 (2018) 2160–2173.
[10] J.F. Canete, R. Baratti, M. Mulas, A. Ruano, Soft-sensing estimation of plant effluent concentrations in a biological wastewater treatment plant using an optimal neural network, Expert Syst. Appl., 63 (2016) 8–19.
[11] M.F. Nezhad, N. Mehrdadi, A. Torabian, S. Behboudian, Artificial neural network modeling of the effluent quality index for municipal wastewater treatment plants using quality variables: south of Tehran wastewater treatment plant, J. Water Supply Res. Technol. AQUA, 65 (2016) 18–27.
[12] M. Bagheri, S.A. Mirbagheri, Z. Bagheri, A.M. Kamarkhanie, Modeling and optimization of activated sludge bulking for a real wastewater treatment plant using hybrid artificial neural networks-genetic algorithm approach, Process Saf. Environ. Prot., 95 (2015) 12–25.
[13] D. Dovžan, V. Logar, I. Škrjanc, Implementation of an evolving fuzzy model (eFuMo) in a monitoring system for a waste-water treatment process, IEEE Trans. Fuzzy Syst., 23 (2015) 1761–1776.
[14] M.M. Ebadzadeh, A.S. Badr, IC-FNN: a novel fuzzy neural network with interpretable intuitive and correlated-contours fuzzy rules for function approximation, IEEE Trans. Fuzzy Syst., 26 (2018) 1288–1302.
[15] J.J. Rubio, SOFMLS: online self-organizing fuzzy modified least-squares network, IEEE Trans. Fuzzy Syst., 17 (2009) 1296–1309.
[16] N. Wang, M.J. Er, X.Y. Meng, A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks, Neurocomputing, 72 (2009) 3818–3829.
[17] J.F. Qiao, W. Li, H.G. Han, Soft computing of biochemical oxygen demand using an improved T–S fuzzy neural network, Chin. J. Chem. Eng., 22 (2014) 1254–1259.
[18] C.F. Juang, C.D. Hsieh, A fuzzy system constructed by rule generation and iterative linear SVR for antecedent and consequent parameter optimization, IEEE Trans. Fuzzy Syst., 20 (2012) 372–384.
[19] M.M. Ebadzadeh, A.S. Badr, CFNN: correlated fuzzy neural network, Neurocomputing, 148 (2015) 430–444.
[20] M.Z. Huang, Y.W. Ma, J.Q. Wan, X.H. Chen, A sensor-software based on a genetic algorithm-based neural fuzzy system for modeling and simulating a wastewater treatment process, Appl. Soft Comput., 27 (2015) 1–10.
[21] L. Teslic, B. Hartmann, O. Nelles, I. Škrjanc, Nonlinear system identification by Gustafson–Kessel fuzzy clustering and supervised local model network learning for the drug absorption spectra process, IEEE Trans. Neural Networks Learn. Syst., 22 (2011) 1941–1951.
[22] H. Malek, M.M. Ebadzadeh, M. Rahmati, Three new fuzzy neural networks learning algorithms based on clustering, training error and genetic algorithm, Appl. Intell., 37 (2012) 280–289.
[23] M. Seera, C.P. Lim, C.K. Loo, H. Singh, A modified fuzzy min–max neural network for data clustering and its application to power quality monitoring, Appl. Soft Comput., 28 (2015) 19–29.
[24] J.J. Tang, F. Liu, W.H. Zhang, R.M. Ke, Y.J. Zou, Lane-changes prediction based on adaptive fuzzy neural network, Expert Syst. Appl., 91 (2018) 452–463.
[25] J.J. Tang, F. Liu, Y.J. Zou, W.B. Zhang, Y.H. Wang, An improved fuzzy neural network for traffic speed prediction considering periodic characteristic, IEEE Trans. Intell. Transp. Syst., 18 (2017) 2340–2350.
[26] J.J. Tang, Y.J. Zou, J. Ash, S. Zhang, F. Liu, Y.H. Wang, Travel time estimation using freeway point detector data based on evolving fuzzy neural inference system, PLoS One, 11 (2016) e0147263.
[27] R.D. Zhang, J.L. Tao, A nonlinear fuzzy neural network modeling approach using an improved genetic algorithm, IEEE Trans. Ind. Electron., 65 (2018) 5882–5892.
[28] Y.Y. Lin, J.Y. Chang, C.T. Lin, Identification and prediction of dynamic systems using an interactively recurrent self-evolving fuzzy neural network, IEEE Trans. Neural Networks Learn Syst., 24 (2013) 310–321.
[29] M. Prasad, C.T. Lin, D.L. Li, C.T. Hong, W.P. Ding, J.Y. Chang, Soft-boosted self-constructing neural fuzzy inference network, IEEE Trans. Syst. Man Cybern. Syst., 47 (2017) 584–588.
[30] S.Q. Wu, M.J. Er, Dynamic fuzzy neural networks-a novel approach to function approximation, IEEE Trans. Syst. Man Cybern. Part B Cybern., 30 (2000) 358–364.
[31] S.Q. Wu, M.J. Er, Y. Gao, A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, IEEE Trans. Fuzzy Syst., 9 (2001) 578–594.
[32] X.J. Ma, J. Yu, Q.L. Dong, A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting, Appl. Soft Comput., 54 (2017) 296–312.
[33] J.C. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy c-means clustering algorithm, Comput. Geosci., 10 (1984) 191–203.

[34] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering and fuzzy clustering: a comparative experimental study, Fuzzy Sets Syst., 161 (2010) 522–543.
[35] M.G. Gong, Y. Liang, J. Shi, W.P. Ma, J.J. Ma, Fuzzy c-means clustering with local information and kernel metric for image segmentation, IEEE Trans. Image Process., 22 (2013) 573–584.
[36] K.P. Lin, A novel evolutionary kernel intuitionistic fuzzy c-means clustering algorithm, IEEE Trans. Fuzzy Syst., 22 (2014) 1074–1087.
[37] B.M. Wilamowski, H. Yu, Improved computation for Levenberg–Marquardt training, IEEE Trans. Neural Networks Learn Syst., 21 (2010) 930–937.
[38] H.G. Han, L.M. Ge, J.F. Qiao, An adaptive second order fuzzy neural network for nonlinear system modeling, Neurocomputing, 214 (2016) 837–847.
[39] Z. Wang, J.S. Chu, Y Song. Y.J. Cui, H. Zhang, X.Q. Zhao, Z.H. Li, J.M. Yao, Influence of operating conditions on the efficiency of domestic wastewater treatment in membrane bioreactors, Desalination, 245 (2009) 73–81.
[40] F.J. Chang, Y.H. Tsai, P.A. Chen, A. Coynel, G. Vachaud, Modeling water quality in an urban river using hydrological factors–data driven approaches, J. Environ. Manage., 151 (2015) 87–96.
[41] E. Lee, S. Han, H. Kim, Development of software sensors for determining total phosphorus and total nitrogen in waters, Int. J. Environ. Res. Public Health, 10 (2013) 219–236.