Research
Environmental Engineering—Article

Development of a Novel Feedforward Neural Network Model Based on Controllable Parameters for Predicting Effluent Total Nitrogen

Zihao Zhao [a], Zihao Wang [a], Jialuo Yuan [a], Jun Ma [a], Zheling He [a], Yilan Xu [d], Xiaojia Shen [e], Liang Zhu [a,b,c,*]

[a] Institution of Environment Pollution Control and Treatment, Department of Environmental Engineering, Zhejiang University, Hangzhou 310058, China
[b] Zhejiang Province Key Laboratory for Water Pollution Control and Environmental Safety, Hangzhou 310058, China
[c] Zhejiang Provincial Engineering Laboratory of Water Pollution Control, Hangzhou 310058, China
[d] Haining Water Investment Group Co., Ltd., Haining 314400, China
[e] Haining Capital Water Co., Ltd., Haining 31440, China

A R T I C L E   I N F O

A B S T R A C T

The problem of effluent total nitrogen (TN) at most of the wastewater treatment plants (WWTPs) in China is important for meeting the related water quality standards, even under the condition of high energy consumption. To achieve better prediction and control of effluent TN concentration, an efficient prediction model, based on controllable operation parameters, was constructed in a sequencing batch reactor process. Compared with previous models, this model has two main characteristics: ① Superficial gas velocity and anoxic time are controllable operation parameters and are selected as the main input parameters instead of dissolved oxygen to improve the model controllability, and ② the model prediction accuracy is improved on the basis of a feedforward neural network (FFNN) with algorithm optimization. The results demonstrated that the FFNN model was efficiently optimized by scaled conjugate gradient, and the performance was excellent compared with other models in terms of the correlation coefficient (R). The optimized FFNN model could provide an accurate prediction of effluent TN based on influent water parameters and key control parameters. This study revealed the possible application of the optimized FFNN model for the efficient removal of pollutants and lower energy consumption at most of the WWTPs.

© 2020 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Due to the recent acceleration of industrialization and urbanization, sewage discharge is increasing in China, reaching 69.96 billion tonnes in 2017 [1]. China has made considerable efforts to decrease the impact of immense sewage discharge, for example, the construction of new wastewater treatment plants (WWTPs), expansion of the treatment capacity of WWTPs, and modification of wastewater discharge standards. At the end of 2018, a total of 5370 WWTPs were operating, with a total treatment capacity of 2.01×10^8 m^3·d^{-1}. The overall electricity consumption reached 1.973×10^{10} kW·h [2]. The existing WWTPs continue to experience several problems such as unstable effluent, high energy consumption, and low-level automation. As one of the most widely used wastewater treatment processes, a sequencing batch reactor (SBR) has the advantages of being a simple process with a flexible operation mode, and excellent influent loading resistance [3]. However, the effluent total nitrogen (TN) concentration of this process is fluctuant, leading to unsatisfactory discharge and energy consumption. According to recent studies [4], the main reason for this problem is the low-level automation of the SBR process.

The TN removal process in the SBR is complex and includes nitrification and denitrification. Moreover, aerobic and anoxic conditions are suitable for nitrification and denitrification, respectively [5]. Dissolved oxygen (DO) is considered to be a significant factor for nitrogen removal process control. During a biological wastewater treatment process, sufficient DO is necessary to ensure the degradation and nitrification of organic matter. Excessive DO leads to high energy consumption, deterioration of sludge flocs, and low-efficiency denitrification [6]. However, the precise control of DO is rarely achieved in the biological wastewater treatment system of WWTPs, and unstable nitrogen removal occurs. According to previous studies, inaccurate monitoring and the slow response of DO caused by unmatched equipment are the main reasons for these shortcomings. In contrast, traditional simulation models and

---

* Corresponding author.
  E-mail address: felix79cn@zju.edu.cn (L. Zhu).

control theory are backward, and the non-timely DO data is another reason for an unstable nitrogen removal process [7,8].

In most WWTPs, engineers usually adjust the DO concentration empirically via controllable parameters such as superficial gas velocity and anoxic time. Compared with the difficulty in the precise control of DO, superficial gas velocity and anoxic time are prone to be controlled, because these parameters provide accurate regulation, precise monitoring, and a fast response. Therefore, an advanced SBR simulation model should be established with superficial gas velocity and anoxic time as the main input parameters instead of DO, and hence, the imprecise control of DO would be effectively avoided. Furthermore, the TN removal performance could be predicted more precisely.

Considering that biological wastewater treatment is a multi-parameter and complex process, traditional biochemistry models are not likely to be applied at an engineering scale because of the poor learning ability for high dimensional and nonlinear data [9,10]. As an effective tool for predicting complicated nonlinear systems, artificial neural network (ANN) is gradually being developed as an appropriate method to simulate the complex biological treatment process in WWTPs. ANN is a self-learning method and has the ability to approximate any nonlinear functions [11–13]. From the perspective of neuron topology, it could be divided into the feedback neural network and feedforward neural network (FFNN), and an FFNN model could theoretically approximate any continuous function with arbitrary precision and possess the strong ability of classification and pattern recognition [14–19]. To improve the prediction ability and efficiency of an FFNN, multifarious optimization algorithms have been proposed, including Levenberg–Marquardt (L–M), Bayesian regularization (BR), scaled conjugate gradient (SCG), momentum, and Nesterov accelerated gradient. All these algorithms could significantly improve the prediction ability and efficiency of an FFNN model [20–22].

In this study, an efficient simulation model was constructed for an SBR process to achieve better prediction and control of effluent TN concentration. Compared with previous studies, the model made significant progress in two main aspects: ① The controllability of TN removal was improved by applying superficial gas velocity and anoxic time as the main input parameters instead of the regular index of DO, and ② the prediction accuracy of the simulation model was improved, based on an optimized FFNN. The objectives of this study are to ① evaluate the prediction performance of effluent TN concentration on an SBR process based on the optimized FFNN model, and ② achieve the accurate control strategy with controllable operation parameters to realize the efficient removal of pollutants and lower energy consumption at most of the WWTPs.

2. Materials and methods

2.1. Long-term simulation of SBR process

Two paralleled reactors (R1, R2) were established to achieve a long-term (two months) biological process simulation of an SBR process. To better simulate the actual situation of wastewater treatment, the reactors were constructed according to the SBR process in the Dingqiao WWTP in Zhejiang, China. The activated sludge in the two reactors was inoculated with seeding sludge from the Dingqiao WWTP. The synthetic wastewater was prepared according to the actual wastewater. The detailed structure of reactors and composition of synthetic wastewater are shown in Fig. S1 and Table S1 in Appendix A.

According to the actual operation mode of the SBR process at the Dingqiao WWTP, each reactor ran for four hours per cycle with a volume exchange ratio of 50%, including feeding period (5 min),

anoxic and aeration period (210 min), setting period (5 min), decanting period (5 min) and idle period (15 min). For the fluctuation of actual influent wastewater quality, the concentration of major influent quality indicators was randomly controlled at 75%–125% of the preset value. Superficial gas velocity and anoxic time were set as control variables with four design values (Table 1). Following standard methods [23], the influent and effluent were sampled periodically to analyze the concentrations of TN, ammonia nitrogen (NH_4^+–N), chemical oxygen demand (COD), and total phosphorous (TP). After the two-month simulation period, 124 groups of data were collected from a total of 16 combinations of control variables, with different values.

To simulate more complicated condition in real SBR process, an extended experiment was constructed with a wider range of controllable variables and different influent ratio after anoxic period (Table 2). A total of 91 groups of data were collected from combinations of these variables, and 11 groups of data were collected from combinations of two superficial gas velocities (3.6 and 4.8 cm·s^{-1}) and two anoxic times (0 and 150 min) to simulate the extreme condition.

2.2. FFNN modeling

The base FFNN model and its optimization algorithm were built and operated in the Matlab R2016a program. Before modeling, the data set, gathered from long-term simulations, were normalized to the range of 0.001–0.999 to eliminate the effect of different dimension. A relatively ideal FFNN model was obtained by constant optimization, and its weight network was adjusted by the error between predicted and actual values (Eq. (1)) [24]. The structure of the neural network consisted of an input layer, a hidden layer, and an output layer. Six variables, including influent water quality (COD, TN, NH_4^+–N, and TP) and operation parameters (superficial gas velocity and anoxic time) were set as the input of the base FFNN model, and the effluent TN was set as the output. The scheme for base FFNN modeling and coding are illustrated in Fig. 1. The dimension of experimental data was relatively low, and therefore, the number of hidden layers was set as one to shorten operation time, improve efficiency, and prevent overfitting. The number of nodes in the hidden layer was calculated by the empirical equation (Eq. (2)).

$$a = Wd + b \tag{1}$$

where a is the output variable; W is the weight matrix; d is the input variable; and b is the matrix of biases in network.

$$h = \sqrt{i + o} + c \tag{2}$$

Table 1
Design values of controllable operation parameters in SBR process.

Level	Superficial gas velocity (cm·s^{-1})	Anoxic time (min)
1	0.5	0
2	1.0	20
3	1.5	40
4	2.0	60

Table 2
Design values of controllable variables and influent ratio after anoxic period in extended experiment.

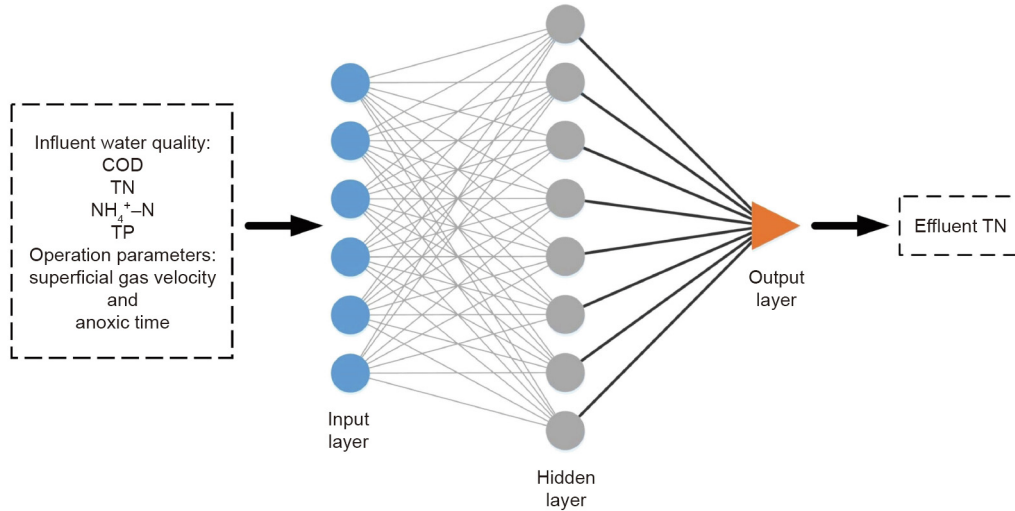| Level	Superficial gas velocity (cm·s^{-1})	Anoxic time (min)	Influent ratio after anoxic period
1	0.4	30	0
2	1.2	60	0.25
3	2.0	90	0.5
4	2.8	120	0.75

Fig. 1. Network structure of FFNN model for effluent TN prediction.

where h is the number of hidden layer nodes; i is the number of input layer nodes; o is the number of output layer nodes; and c is the adjustment constant between 1 and 10.

2.3. Optimization of the FFNN model

The first base FFNN model constructed was the back propagation (BP) neural network. In the training period of the BP neural network, the error between predicted and actual values could propagate back to the hidden layer. According to error backward propagation, the BP neural network could adjust the network weight constantly until the error was minimized. The gradient descent algorithm was the most common method applied to adjust the weights continuously. It could adjust the network weight in the direction of gradient descent and the minimal error could finally be obtained [25]. In the actual training process, the gradient descent method is prone to fall into the local minimum value instead of the global minimum value, which thus, reduces the learning efficiency and prediction accuracy [26]. To improve the learning efficiency and prediction accuracy, three optimization algorithms (L–M, BR, and SCG) were used to optimize the FFNN model, and the best optimization algorithm was selected to obtain the most suitable FFNN model of the SBR process. To improve the model's generalization ability, more complicated data set includes the data collected from the extended experiment, was used to train the FFNN model. The influent ratio after anoxic period was set as the input instead of TP which is thought to have a little effect on total nitrogen removal.

2.3.1. L–M algorithm

The L–M algorithm incorporates the merits of the Gauss–Newton (G–N) algorithm and gradient descent algorithm. Compared with a traditional gradient descent method, it could effectively avoid the local minimum and improve the convergence speed to the global minimum. The algorithm's detail is as follows:

Vector $\mathbf{W}$ is used to represent the weights between layers in the BP neural network. The square sum of error (E) is

$$E = \frac{1}{2}\sum\sum \left(t_{nj} - O_{nj}\right)^2 = \frac{1}{2}\sum {\boldsymbol{\varepsilon}_n}^2 = \frac{1}{2}\|\boldsymbol{\varepsilon}\|^2 \tag{3}$$

where n is the sample number; t_{nj} is the expected output of sample n at node j of the output layer; O_{nj} is the actual output; ε_n is a member of vector $\boldsymbol{\varepsilon}$. In Eq.(4), k is the number of iteration (the number of weight adjustment). In the process of calculating $\mathbf{W}^{k+1}$, if

$\mathbf{W}^{k+1} - \mathbf{W}^k$ is small, $\boldsymbol{\varepsilon}$ could be expanded into a first-order Taylor series:

$$\boldsymbol{\varepsilon}\mathbf{W}^{k+1} = \boldsymbol{\varepsilon}\mathbf{W}^{k+1} + \mathbf{Z}\left(\mathbf{W}^{k+1} - \mathbf{W}^k\right) \tag{4}$$

where $\mathbf{Z}$ is the Jacobian matrix of $\boldsymbol{\varepsilon}$, and the element of $\mathbf{Z}$ is

$$\mathbf{Z}_{nj} = \frac{\partial \boldsymbol{\varepsilon}_n}{\partial \mathbf{W}_j} \tag{5}$$

Therefore, the error function could be changed into

$$E = \frac{1}{2}\left\|\boldsymbol{\varepsilon}\mathbf{W}^k + \mathbf{Z}\left(\mathbf{W}^{k+1} - \mathbf{W}^k\right)\right\|^2 \tag{6}$$

To minimize the error function, $\mathbf{W}^{k+1}$ could be differentiated to obtain the following G–N iterative formula:

$$\mathbf{W}^{k+1} = \mathbf{W}^k - \left(\mathbf{Z}^T\mathbf{Z}\right)^{-1}\mathbf{Z}^T\boldsymbol{\varepsilon}\mathbf{W}^k \tag{7}$$

The error function is rewritten as follows to avoid the Jacobian matrix singularity, which often occurs in the G–N method:

$$E = \frac{1}{2}\left\|\boldsymbol{\varepsilon}\mathbf{W}^k + \mathbf{Z}\left(\mathbf{W}^{k+1} - \mathbf{W}^k\right)\right\|^2 + \lambda\left\|\mathbf{W}^{k+1} - \mathbf{W}^k\right\|^2 \tag{8}$$

where λ is the damping parameter.

Take the derivative of E, then the L–M iterative formula based on the G–N method could be obtained:

$$\mathbf{W}^{k+1} = \mathbf{W}^k - \left(\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I}\right)^{-1}\mathbf{Z}^T\boldsymbol{\varepsilon}\mathbf{W}^k \tag{9}$$

where $\mathbf{I}$ is the identity matrix and iterative variable. The search direction and training step size are affected by λ during the iteration. When λ is larger at the initial stage of calculation $\mathbf{Z}^T\mathbf{Z}$ is negligible compared to $\lambda\mathbf{I}$, and Eq. (9) could be written as

$$\mathbf{W}^{k+1} = \mathbf{W}^k - \frac{1}{\lambda}\mathbf{g} \tag{10}$$

where $\mathbf{g}$ is the gradient. λ in the expression tends to 0 at the extremum of the function, meaning that it turns into the G–N interval formula.

2.3.2. BR algorithm

BR could regularize the neural network by the Bayes method. The regularization refers to restricting the complexity of the network in the training period by adding a penalty term. After regularization, the overfitting phenomenon could be effectively

197

avoided to improve generalization ability. In general, the performance function (F) of neural network is

$$F = E \tag{11}$$

After adding a penalty term $E_{\boldsymbol{W}}$, the performance function turns into

$$F = \alpha E_{\boldsymbol{W}} + \beta E \tag{12}$$

$$E_{\boldsymbol{W}} = \frac{1}{2}\|\boldsymbol{W}\|^2 \tag{13}$$

The relative size of α and β determines the proportion of the penalty term. If $\alpha \ll \beta$, the performance function is approximate to no regularization, meaning it minimizes the training error but maximizes the possibility of overfitting. If $\alpha \gg \beta$, it focuses on the limitation of the network, making the model's prediction capacity poor. Therefore, knowing how to determine the values of α and β is necessary. In the framework of Bayesian analysis, MacKay deduces that [27]

$$\alpha = \gamma/(2E_{\boldsymbol{W}}) \tag{14}$$

$$\beta = \frac{N - \gamma}{2E} \tag{15}$$

where $\gamma = N - 2\alpha \mathrm{tr}(\boldsymbol{H})^{-1}$, represents the valid weight and N is the total number of samples. $\boldsymbol{H}$ is the Hessian matrix of F:

$$\boldsymbol{H} = \nabla^2 \alpha E_{\boldsymbol{W}} + \nabla^2 E \tag{16}$$

The Hessian matrix is very computationally intensive, and Foresee F. Dan and Martin T. Hagan approximated the Hessian matrix with the G–N method, which greatly reduces the amount of computation [28]:

$$\boldsymbol{H} = \nabla^2 F \approx 2\beta \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + 2\alpha \boldsymbol{I}_N \tag{17}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the training error.

2.3.3. SCG algorithm

The SCG is an improved learning algorithm of standard BP. In the traditional gradient descent method, a direction of gradient descent is perpendicular to the previous direction, causing the global minimum difficult to approximate. The conjugate gradient method determined the new search direction by combining the new direction with the previous search direction. The conjugate gradient method is computationally intensive, requiring a new search in each iteration. The SCG proposed by Moller successfully avoids this drawback by adding a confidence interval to the conjugate gradient [29]. The adjustment method of the SCG algorithm is

$$w_{k+1} = w_k + \theta_k \boldsymbol{p}_k \tag{18}$$

where w_k is a point in $\boldsymbol{W}_k$, $\boldsymbol{p}_k$ is the search direction, and θ_k is the search step size of iteration k.

$$\theta_k = -\frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{p}_k}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{H}_k\boldsymbol{p}_k} \tag{19}$$

where $\boldsymbol{g}_k$ is the current gradient of the function, and $\boldsymbol{H}_k$ is the Hessian matrix of iteration k.

Set $\boldsymbol{s}_k = \boldsymbol{H}_k\boldsymbol{p}_k$, $\delta_k = \boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{s}_k$, $u_k = -\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{p}_k$, then $\theta_k = u_k/\delta_k$. In order to ensure $\delta_k > 0$, set $\delta_k = \boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{s}_k + \lambda_k|\boldsymbol{p}_k|^2$.

The step size θ_k can then be determined:

$$\theta_k = \frac{u_k}{\delta_k} = \frac{u_k}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{s}_k + \lambda_k|\boldsymbol{p}_k|^2} \tag{20}$$

2.4. Verification and comparison of different models

The predictive performance of each model was assessed by the correlation coefficient (R) (Eq. (21)), and the training performance

of each model was compared by the mean square error (MSE) (Eq. (22)). The R and MSE were calculated in the Matlab R2016a program.

$$R = \frac{\mathrm{Cov}(\mathrm{TN_P}, \mathrm{TN_A})}{\sigma_\mathrm{P}\sigma_\mathrm{A}} \tag{21}$$

$$\mathrm{MSE} = \frac{1}{m}\sum_{n=1}^{m}(\mathrm{TN}_{\mathrm{P}n} - \mathrm{TN}_{\mathrm{A}n})^2 \tag{22}$$

where $\mathrm{Cov}(\mathrm{TN_P}, \mathrm{TN_A})$ is the covariance of predicted effluent TN and actual effluent TN; σ_P and σ_A are the standard deviation of predicted effluent TN and actual effluent TN, respectively; m is the number of the datasets; $\mathrm{TN}_{\mathrm{P}n}$ and $\mathrm{TN}_{\mathrm{A}n}$ are the predicted effluent TN and actual effluent TN for sample n.

3. Results and discussion

3.1. Feasibility of the FFNN model using actual operation parameters as main input variables

The number of nodes in the hidden layer needs to be determined prior to application. With other parameters constant, the number of hidden layer nodes was selected as 3 to 12 for training, according to Eq. (2). Fig. 2 shows the MSE for the different number of hidden layer nodes after training. By comparison, the number of hidden layer nodes was selected as 8, and therefore, the base FFNN model's structure was constructed with 6–8–1 (Fig. 1). Consequently, MSE obtained the lowest value.

To train and assess the base FFNN model, the data collected from the long-term laboratory simulation of the SBR were randomly divided into a training set (104 groups) and a testing set (20 groups). Fig. 3 indicates a consistent trend overall, and the predicted value on the left side (training set) was close to the actual value (R = 0.91973), indicating that it was feasible to build a simulation model by replacing DO with actual controllable operation parameters such as superficial gas velocity and anoxic time as the FFNN input.

The prediction curve of the testing set on the right was, however, far from the actual situation (R = 0.5057), indicating that the simulation model still had some problems such as overfitting or falling into local minimum value which could reduce the prediction performance. To solve these issues, three algorithms (L–M, BR, and SCG) were used to optimize the FFNN model. A more complicated data set was used to train the model, and more reasonable training mode for different algorithms was established to find the best algorithm for the model.
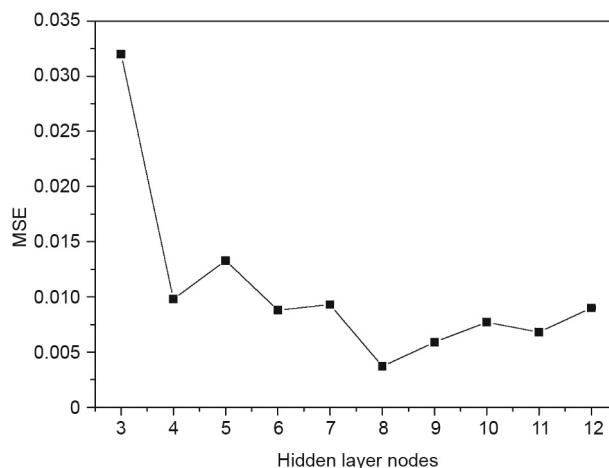

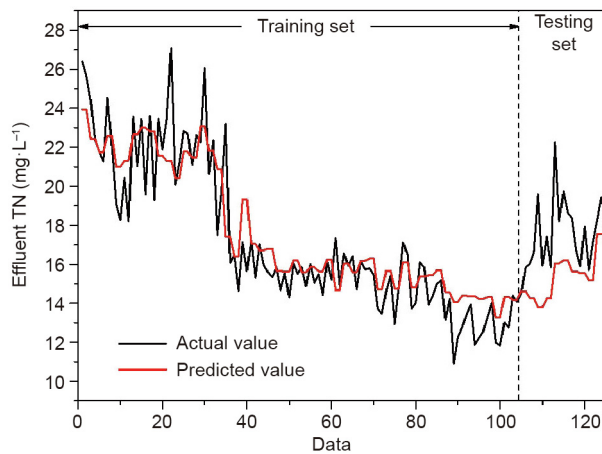**Fig. 2.** MSE for FFNN with different hidden layer nodes.

Fig. 3. Comparison of predicted results and actual values of FFNN.

3.2. Optimization of the FFNN model by L–M algorithm

The data set were randomly divided into a training set (80%) and a validation set (20%). After optimization by trial method, the number of hidden layer nodes was set as 30, the minimum number of failure times was set as 100, and tansig was selected as the transfer function between the hidden layer and the output layer. After training (113 steps), the training stopped because the number of failure times exceeded the preset value, with a shared time of 1.0 s. The simulation results are shown in Fig. 4. The MSE was 0.000200, and the R value in the training set, the validation set, and the whole set were 0.93603, 0.92316, and 0.93392, respectively, illustrating a good performance in effluent TN prediction. Given the higher R value and faster convergence speed than previous model, the model optimized by the L–M algorithm not only avoided the local minimum effectively, but also improved the convergence speed to the global minimum.

3.3. Optimization of the FFNN model by BR algorithm

The BR algorithm did not need a validation set, and therefore, the data were randomly divided into a training set (80%), and a testing set (20%). In the process of model debugging, it was found that when hidden layer nodes were less than 20, the R value was lower than 0.6, and the results indicated that the prediction ability was poor. With the number of hidden layer nodes increased to 30, the R value gradually improved. After the number of nodes

exceeded 30, the R value increased insignificantly, whereas the training time extended significantly. Consequently, the number of hidden layer nodes was selected as 30. Tansig was selected as the transfer function, the upper limit of training steps was set to 1000, and the training time was 19 s. As shown in Fig. 5, the R value of the training set is 0.90232, whereas the R value of the testing set is 0.83000. It was determined that the performance of the model was better than that of the base model and worse than that of the model optimized by the L–M algorithm. The BR algorithm improved the model performance by restricting the complexity of the network, leading to a higher R value than the base model. Nonetheless, several relationships between input and output may be lost due to the regularization, particularly in the case whereby the complexity of the model's structure was moderate.

3.4. Optimization of the FFNN model by SCG algorithm

Similar to the L–M algorithm, the data set was divided into a training set (80%) and a validation set (20%). The number of hidden layer nodes was maintained at 30 to ensure the identical model structure with the three algorithms. Tansig was selected as the transfer function between the hidden layer and the output layer, and the minimum failure time was 100. The training stopped after 156 steps because the number of failure times exceeded the preset value, and the usage time was near to zero. As shown in Fig. 6, the R value based on the training set and the validation set were 0.91653 and 0.96276, respectively. The higher R value of validation set revealed an inexistence of the overfitting phenomenon. With these results, it was concluded that the prediction performance of the FFNN model was excellent.

3.5. Comparison of three optimization algorithms

The relevant training parameters of FFNN, based on different optimization algorithms, are summarized in Table 3. The results showed that in the training period, all three FFNN models with optimization algorithms had satisfactory MSE (< 0.001), and all gave better performances than the model without the algorithm optimization (MSE = 0.0037). Among them, the SCG algorithm was characterized by the shortest training time because of the rapid training method. The R values of FFNN, based on different optimization algorithms, are summarized in Table 4. In the training period, the three models obtained R values of 0.93603, 0.90232, and 0.91653, respectively, which was similar to the base FFNN model. The results revealed the intense relationship between input
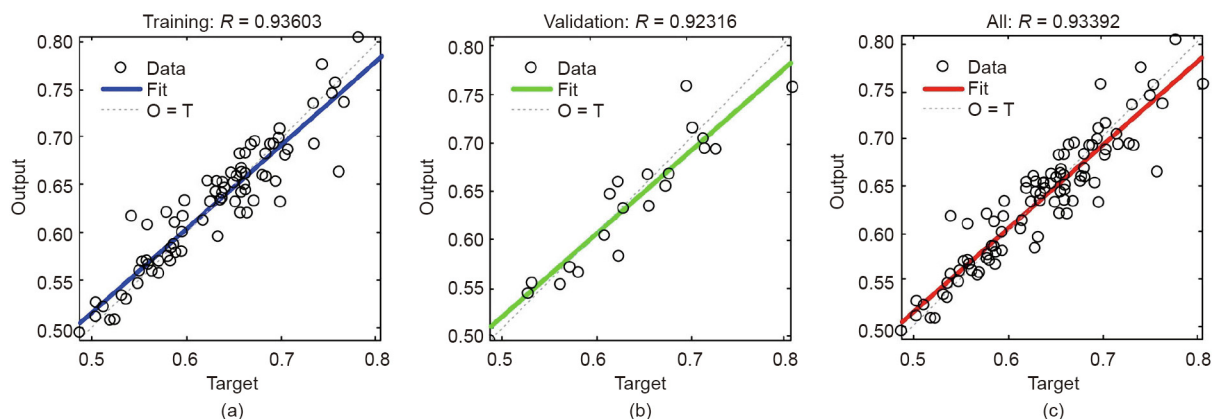

Fig. 4. FFNN simulating results based on L–M algorithm: (a) the training set (the output ≈ 0.88 × target + 0.075); (b) the validation set (the output ≈ 0.88 × target + 0.074); and (c) the whole set (the output ≈ 0.88 × target + 0.077). The line "Fit" refers to the relationship between the target and the output and the line "O = T" means the target is equal to the output.
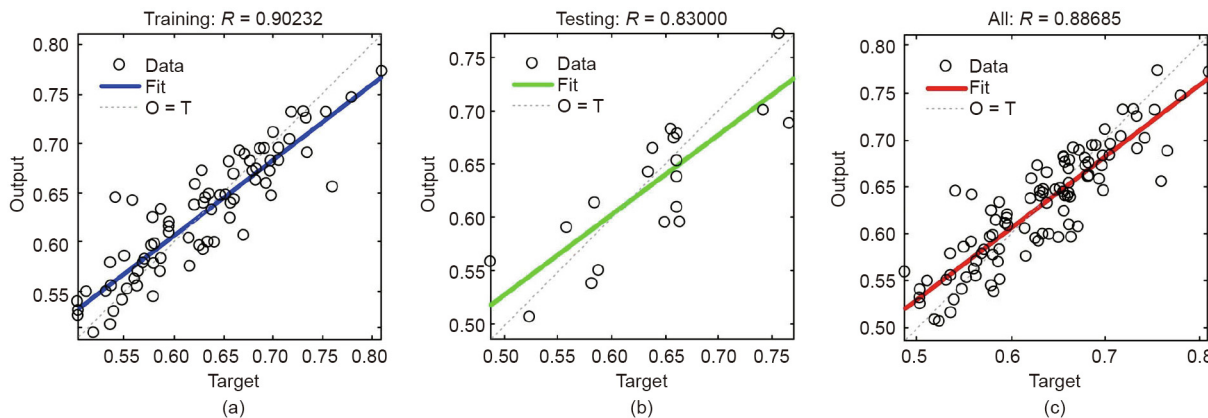
Fig. 5. FFNN simulating results based on BR algorithm: (a) the training set (the output ≈ 0.76 × target + 0.15); (b) the testing set (the output ≈ 0.75 × target + 0.15); and (c) the whole set (the output ≈ 0.76 × target + 0.15).

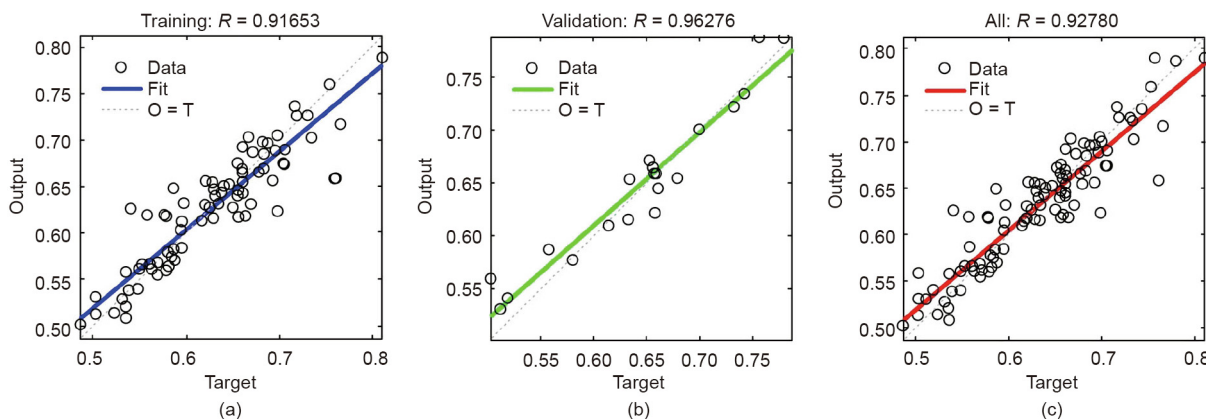**Fig. 6.** FFNN training results based on SCG algorithm: (a) the training set (the output ≈ 0.84 × target + 0.1); (b) the validation set (the output ≈ 0.88 × target + 0.079); and (c) the whole set (the output ≈ 0.85 × target + 0.092).

Table 3
Relevant training parameters of FFNN based on different optimization algorithms.

Optimization algorithm	Training step	Training time (s)	MSE
L–M	113	1	0.000200
BR	1000	19	0.000935
SCG	156	≪ 1	0.000512

Table 4
R values in different datasets of FFNN based on different optimization algorithms.

Dataset	L–M	BR	SCG
Training set	0.93603	0.90232	0.91653
Validation set	0.92316	—	0.96276
Testing set	—	0.83000	—
The whole set	0.93392	0.88685	0.92780

(influent water parameters and operation parameters) and output (effluent TN) from the training set. In the prediction period (testing and validation), all of the FFNN models obtained a higher R value than the base FFNN model. The results indicated that the prediction performance of the FFNN models for effluent TN was greatly improved by the optimization algorithms. The R value of the validation set of SCG was the highest in the three algorithms, and higher than the R value of the training set. The results

demonstrated that the SCG algorithm obtained the most rapid calculation speed and outstanding prediction ability, and was an excellent algorithm for optimizing the FFNN model.

3.6. Comparisons with existing prediction models

Diverse simulation models have been proposed to achieve the prediction of effluent water quality in different wastewater treatment processes such as anaerobic–anoxic–oxic (A^2/O), SBR, and oxidation ditch with different model methods (traditional machine learning methods and ANN), and all the models were well suited to the specific environment. To predict NH_4^+–N removal of slaughterhouse wastewater, a BP neural network model based on influent water quality and DO was proposed by Kundu et al. [30]. TN removal prediction of aerobic granulation was achieved using the ANN model based on influent water parameters [31]. A study by Ebrahimi et al. [32] proposed a multivariate regression model which could predict the effluent quality (biochemical oxygen demand (BOD), TP) of oxidation ditch. Nevertheless, the parameters selected in this study were mainly focused on DO and influent water parameters, but not controllability. The regulation strategies based on the models was lacking, because the feedback from the model was delayed.

Compared with previous studies, strong controllability and high prediction accuracy of simulation models were the two aspects highlighted in this study. This study is the first to demonstrate that

Table 5
Performance comparison of different prediction methods (R values are rounded to two decimal points).

Method	MSE	R
FFNN	0.000512	0.93
LR	0.009622	0.79
Lasso	0.009696	0.79
Ridge	0.009299	0.80
SVR	0.007687	0.84
RF	0.006941	0.85

TN removal performance could be controlled by the FFNN model based on actual controllable operation parameters such as superficial gas velocity and anoxic time, instead of DO, when considering the uncontrollability of DO. For the improvement of prediction accuracy, various algorithms were applied to optimize the FFNN model, and the model optimized by SCG was considered to have the highest prediction accuracy. To verify the advantage of the FFNN model optimized by SCG for effluent TN prediction, the FFNN model and other existing modeling methods, such as linear regression (LR), ridge regression, least absolute shrinkage and selection operator (Lasso) regression, support vector regression (SVR), and random forest (RF) were conducted (Table 5). The results showed that FFNN optimized by SCG was the best modeling method in this study, considering the lowest MSE and the highest R value of this model. It is reasonable that FFNN, optimized by SCG, could provide an accurate prediction of effluent TN, and fitted the real SBR process well.

In the future, the application of this prediction model could achieve the regulation of controllable operation parameters continuously, based on influent water parameters. Therefore, the effluent TN concentration could be maintained at a stable level and meet the relevant water quality standards. Furthermore, excessive aeration could be avoided effectively via continuous regulation of the operation parameters [33]. Finally, the energy consumption of controllable operation parameters would be minimized.

4. Conclusions

In this study, a SBR simulation model was developed, and the suitable control of effluent TN concentration and energy consumption were achieved after a long-term simulation of the SBR process. Superficial gas velocity and anoxic time were selected as the main input parameters of the FFNN model instead of DO, to improve the controllability of the model. Furthermore, SCG was selected as the algorithm for the optimization of the FFNN model because of the fastest calculation speed and outstanding prediction ability. Finally, the optimized FFNN model exhibited an excellent prediction capacity of effluent TN and optimal operation parameters. The findings of this study could provide a valuable solution for the stable operation and energy saving of WWTPs according to influent quality and effluent standards.

Acknowledgements

Compliance with ethics guidelines

Zihao Zhao, Zihao Wang, Jialuo Yuan, Jun Ma, Zheling He, Yilan Xu, Xiaojia Shen, and Liang Zhu declare that they have no conflict of interest or financial conflicts to disclose.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eng.2020.07.027.

References

[1] National Bureau of Statistics of China. China statistical yearbook 2019. Beijing: China Statistics Press; 2019. Chinese.
[2] Lu J, Wang X, Liu H, Li W. Optimizing operation of municipal wastewater treatment plants in China: the remaining barriers and future implications. Environ Int 2019;129:273–8.
[3] Irvine RL, Ketchum LH Jr, Asano T. Sequencing batch reactors for biological wastewater treatment. Crit Rev Environ Control 1989;18(4):255–94.
[4] Vieira M, Brito AG, Nogueira R. Nitrogen removal in a sequencing batch biofilm reactor: effect of carbon availability and intermittent aeration. World Rev Sci Techonol Sustainable Dev 2009;6(2–4):173.
[5] Ren Y, Ngo HH, Guo W, Wang D, Peng L, Ni BJ, et al. New perspectives on microbial communities and biological nitrogen removal processes in wastewater treatment systems. Bioresour Technol 2020;297:122491.
[6] Hu L, Wang J, Wen X, Qian Y. Study on performance characteristics of SBR under limited dissolved oxygen. Process Biochem 2005;40(1):293–6.
[7] Boon AG. Sequencing batch reactors: a review. Water Environ J 2003;17(2):68–73.
[8] Singh M, Srivastava RK. Sequencing batch reactor technology for biological wastewater treatment: a review. Asia-Pac J Chem Eng 2011;6(1):3–13.
[9] Ingimundarson A, Ocampo-Martinez C, Bemporad A. Suboptimal model predictive control of hybrid systems based on mode-switching constraints. In: Proceedings of the 2007 46th IEEE Conference on Decision and Control; 2007 Dec 12–14; New Orleans, LA, USA. New York: IEEE; 2008.
[10] Olsson G, Newell B. Wastewater treatment systems: modelling, diagnosis and control. London: IWA Publishing; 2015.
[11] Baxter CW, Zhang Q, Stanley SJ, Shariff R, Tupas R-RT, Stark HL. Drinking water quality and treatment: the use of artificial neural networks. Can J Civ Eng 2001;28(S1):26–35.
[12] Miron M, Frangu L, Ifrim G, Garaman S. Modeling of a wastewater treatment process using neural networks. In: Proceedings of the 2016 20th International Conference on System Theory, Control and Computing; 2016 Oct 13–15; Sinaia, Romania. New York: IEEE; 2016.
[13] Fan M, Li T, Hu J, Cao R, Wu Q, Wei X, et al. Synthesis and characterization of reduced graphene oxide-supported nanoscale zero-valent iron (nZVI/rGO) composites used for Pb(II) removal. Materials 2016;9(8):687.
[14] Dreyfus G. Neural networks: an overview. In: Dreyfus G, editor. Neural networks—methodology and applications. Heidelberg: Springer; 2005. p. 1–83.
[15] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2(5):359–66.
[16] Ata R. Artificial neural networks applications in wind energy systems: a review. Renew Sustain Energy Rev 2018;84:173. Retraction of: Ata R. Renew Sustain Energy Rev 2015;49:534–62.
[17] Almási A, Woźniak S, Cristea V, Leblebici Y, Engbersen T. Review of advances in neural networks: neural design technology stack. Neurocomputing 2016;174(Pt A):31–41.
[18] Lin D, Vasilakos AV, Tang Y, Yao Y. Neural networks for computer-aided diagnosis in medicine: a review. Neurocomputing 2016;216:700–8.
[19] Lesnik KL, Liu H. Predicting microbial fuel cell biofilm communities and bioreactor performance using artificial neural networks. Environ Sci Technol 2017;51(18):10881–92.
[20] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. Nature 2017;550(7676):354–9.
[21] Qian N. On the momentum term in gradient descent learning algorithms. Neural Netw 1999;12(1):145–51.
[22] Sadrzadeh M, Mohammadi T, Ivakpour J, Kasiri N. Neural network modeling of Pb^{2+} removal from wastewater using electrodialysis. Chem Eng Process 2009;48(8):1371–81.
[23] Association APH. Standard methods for the examination of water and wastewater. 21st ed. Washington, DC: American Public Health Association; 2005.
[24] Ojha VK, Abraham A, Snášel V. Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng Appl Artif Intel 2017;60:97–116.
[25] Anzai Y. Learning by neural networks. In: Anzai Y, editor. Pattern recognition & machine learning. San Francisco: Morgan Kaufmann; 1992. p. 297–335.
[26] Zhang S, Choromanska A, LeCun Y. Deep learning with elastic averaging SGD. 2014. arXiv:1412.6651.
[27] MacKay DJC. A practical bayesian framework for backpropagation networks. Neural Comput 1992;4(3):448–72.
[28] Foresee F, Hagan M. Gauss–Newton approximation to Bayesian regularization. In: Proceedings of International-joint Conference on Neural Networks; 1997 Jun 9–12; Houston, TX, USA; New York: IEEE; 1997.
[29] Moller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 1993;6(4):525–33.

[30] Kundu P, Debsarkar A, Mukherjee S. Artificial neural network modeling for biological removal of organic carbon and nitrogen from slaughterhouse wastewater in a sequencing batch reactor. Adv Artif Neural Syst 2013;2013:268064.

[31] Gong H, Pishgar R, Tay JH. Artificial neural network modelling for organic and total nitrogen removal of aerobic granulation under steady-state condition. Environ Technol 2019;40(24):3124–39.

[32] Ebrahimi M, Gerber EL, Rockaway TD. Temporal performance assessment of wastewater treatment plants by using multivariate statistical analysis. J Environ Manage 2017;193:234–46.

[33] Brdjanovic D, Slamet A, van Loosdrecht MCM, Hooijmans CM, Alaerts GJ, Heijnen JJ. Impact of excessive aeration on biological phosphorus removal from wastewater. Water Res 1998;32(1):200–8.