Environmental Science and Ecotechnology 13 (2023) 100207

Contents lists available at ScienceDirect

Environmental Science and Ecotechnology

journal homepage: www.journals.elsevier.com/environmental-science-and-ecotechnology/

Original Research

Deep learning-based prediction of effluent quality of a constructed wetland

Bowen Yang [a], Zijie Xiao [a], Qingjie Meng [b], Yuan Yuan [c], Wenqian Wang [a], Haoyu Wang [d], Yongmei Wang [a], Xiaochi Feng [a, *]

[a] State Key Laboratory of Urban Water Resource and Environment, School of Civil and Environmental Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, Guangdong, 518055, PR China
[b] Shenzhen Shenshui Water Resources Consulting CO, LTD, Shenzhen, Guangdong, 518022, PR China
[c] College of Biological Engineering, Beijing Polytechnic, Beijing, 10076, PR China
[d] State Environmental Protection Key Laboratory of Integrated Surface Water-Groundwater Pollution Control, School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen, 518055, China

ARTICLE INFO

Article history:
Received 31 May 2022
Received in revised form
16 September 2022
Accepted 16 September 2022

Keywords:
LSTM
Constructed wetlands
Water quality prediction
Deep learning
Multi-source data fusion

ABSTRACT

Data-driven approaches that make timely predictions about pollutant concentrations in the effluent of constructed wetlands are essential for improving the treatment performance of constructed wetlands. However, the effect of the meteorological condition and flow changes in a real scenario are generally neglected in water quality prediction. To address this problem, in this study, we propose an approach based on multi-source data fusion that considers the following indicators: water quality indicators, water quantity indicators, and meteorological indicators. In this study, we establish four representative methods to simultaneously predict the concentrations of three representative pollutants in the effluent of a practical large-scale constructed wetland: (1) multiple linear regression; (2) backpropagation neural network (BPNN); (3) genetic algorithm combined with the BPNN to solve the local minima problem; and (4) long short-term memory (LSTM) neural network to consider the influence of past results on the present. The results suggest that the LSTM-predicting model performed considerably better than the other deep neural network-based model or linear method, with a satisfactory R^2. Additionally, given the huge fluctuation of different pollutant concentrations in the effluent, we used a moving average method to smooth the original data, which successfully improved the accuracy of traditional neural networks and hybrid neural networks. The results of this study indicate that the hybrid modeling concept that combines intelligent and scientific data preprocessing methods with deep learning algorithms is a feasible approach for forecasting water quality in the effluent of actual engineering.

1. Introduction

Compared with wastewater treatment plants, constructed wetlands (CWs) are widely applied in developing countries to deeply purify urban water pollution because of their low construction and operation costs, excellent treatment capacity and high ecological benefits [1,2]. Additionally, in the context of global warming, new requirement has been presented for wastewater treatment, that is, the reduction of greenhouse gas (GHG) emissions [3,4]. In this case, CWs are widely used as a low-carbon and green sewage treatment method to address various point and non-point source pollution [5]. To maximize the treatment efficiency of CWs, it is necessary to make timely predictions about the potential changes in effluent and adjust the operation parameters of CWs to guarantee the safety of urban water systems [6]. Therefore, based on the optimization of previous effluent quality data from a CW, establishing a satisfactory model to predict sudden future changes will provide an effective strategy for the regulation of CWs, thereby indirectly providing an approach to control urban water pollution [7–9].

Mathematical models have been used frequently to not only simulate CW purification mechanisms but also predict effluent quality [10,11]. However, to predict the effluent quality of CWs

* Corresponding author. Harbin Institute of Technology (Shenzhen), China.
E-mail address: fengxiaochi@hit.edu.cn (X. Feng).

https://doi.org/10.1016/j.ese.2022.100207

based on mathematical models, it is not only necessary to continuously monitor a series of key water quality indicators (biochemical oxygen demand in five days (BOD_5), chemical oxygen demand (COD), ammonia nitrogen (NH_4^+-N), and total phosphorus (TP)) but also to measure the absorption of wetland plants and the activity of bacteria, which consumes a large amount of time and energy [12,13]. For example [14], established a physical-mathematical water quality model to simulate the interaction between overland and subsurface flow that occurs in horizontal flow CWs. The process not only required a series of specific formulas to simulate biochemical processes but also needed to establish a water hydraulic model, which was extremely tedious. Therefore, time-consuming sampling and measurement was a major obstacle in water quality perception and the timely adjustment of CWs.

Meanwhile, various data-driven models have been used to predict the purification capacity of CWs [15]. Although a model requires a number of data points as an mechanistic or mathematical model, the data-driven method does not require detailed fundamental and mechanistic knowledge. Therefore, data-driven models have the potential for wider application and achieve better prediction performance in terms of forecasting the water quality of practically CWs than mathematical models [16,17].

Among the diverse data-driven methods, deep learning has become a widely used technology in hydrological time series prediction because of its strong nonlinear mapping and prediction capabilities, higher error tolerance and better generalizability [18]. For example [19], optimized energy consumption and effluent quality during wastewater treatment using novel dynamic optimization control based on multi-objective ant lion optimization and deep learning algorithms [20]. applied an artificial neural network (ANN) to simulate the denitrification rate of CWs and concluded that the ANN achieved a much better simulation effect than the traditional multiple linear regression (MLR) model or simplified mechanistic model because of its excellent regression capabilities for nonlinear problems [21]. used a genetic algorithm (GA) combined with an ANN model to simulate and predict papermaking wastewater treatment. The results demonstrated that, through its excellent global searchability, the GA can substantially reduce the BPNN's error and improve accuracy, which makes it a powerful tool for predicting complex problems [22]. Additionally [23], used a long short-term memory (LSTM) model combined with the wavelet domain threshold denoising method to predict historical changes in chlorophyll A in lake water and predict future concentration changes. Furthermore [24], proposed an integrated empirical mode decomposition (EMD)-LSTM model to predict water quality in urban drainage networks, which combined an EMD-centric data preprocessing module and LSTM neural network prediction module to improve the model-based accuracy of the detection method. These results demonstrated that LSTM performed well in multi-time-step prediction problems.

To date, the large-scale application of deep learning methods for predicting effluent quality in real vertical flow CWs has not been investigated systematically. Previous applications have either been in small-scale CWs in the laboratory or mostly focused on predicting the concentration of specific pollutants based on several accessible parameters, such as temperature, flow rate, and dissolved oxygen [25–29]. However, considering that the water influent concentration of CWs under actual conditions is highly volatile and that a large number of parameters affect the processing capacity of CWs in large-scale applications, such as temperature, rainfall, atmospheric pressure, and humidity, it remains a challenge to establish a suitable method to predict multiple pollutants simultaneously with the help of multi-source data.

Therefore, our purpose in this study is to simulate and predict the effluent quality of large-scale CWs in time through a combination of deep learning algorithms and multi-source data-driven methods. First, given the multi-source data that affect the processing capacity of CWs, we investigate the mapping relationship between the data of the previous day and the concentration of pollutants in the CW effluent of the next day. Then, we develop various typical approaches for predicting the concentrations of three conventional pollutants and compare them with each other so that we can identify the best model for this complex environment at large spatial scales. Finally, because of the high volatility of the effluent concentration of CWs, we propose a data preprocessing module that can smooth the original data, remove high-frequency noise, and effectively increase model prediction accuracy. Our research provides new methods and ideas for improving the prediction accuracy of the large-scale application of water quality models in practical scenarios.

2. Materials and methods

2.1. Preprocessing of raw data

In this study, we divide data preprocessing methods into two parts: moving average and normalization. The moving average is a data smoothing method that is capable of smoothing high-frequency noise, and making the pattern more visible than original is required to ensure the stability of model performance [30]. The smoothing formula is shown in Equation (1). Because of the difference in dimensions between the indicators, some indicators are ignored in the modeling process, and the original variables are normalized through a linear transformation of the raw data (Zhou 2020). For example, if there are i indicators, v_1, v_2, …, v_i, that represent the attributes of j objects, then the raw dataset is as shown in Equation (2). “Min” and “max” are the minimum and maximum values of an index, respectively. These values map the original value v_{ij} of an index to the value v'_{ij} in the interval [0, 1] through min-max normalization, as shown in Equation (3):

$$Y_t = \frac{X_t + X_{t-1} + X_{t-2} + \ldots + X_{t-n}}{n} \tag{1}$$

where X_t is the effluent concentration on day t, Y_t is the effluent concentration on day t after averaging, and n is the average number of days;

$$\boldsymbol{V}_{i\times j} = \begin{pmatrix} V_{11} & \cdots & V_{1j} \\ \vdots & \ddots & \vdots \\ V_{i1} & \cdots & V_{ij} \end{pmatrix}, \tag{2}$$

where i represents the number of indicators and j represents the number of attributes of each indicator; and

$$V'_m = \frac{V_m - \min(V_m)}{\max(V_m) - \min(V_m)} \tag{3}$$

where V'_m represents the normalized value, and $\max(V_m)$ and $\min(V_m)$ are the maximum and minimum values of the sample, respectively.

2.2. Prediction models

2.2.1. Multiple linear regression (MLR)

In regression analysis, if more than one independent variable (input variables x_j) are used to predict dependent variables (output variable Y) through linear regression, this is called MLR [31], which can be expressed as follows:

$$Y = k_1x_1 + k_2x_2 + k_3x_3 + \ldots\ldots + k_jx_j + k_0 \quad (4)$$

where k_1, k_2, …, k_n are the regression coefficients and k_0 is the intercept of MLR. The coefficients of each variable reflect its effect on the predictive results.

Multicollinearity is a common problem in MLR. When there is strong collinearity between variables, the prediction performance of the model decreases. Therefore, it is necessary to calculate the variance inflation factor (VIF) value between the variables. The VIF value of each independent variable is calculated as

$$\mathrm{VIF} = \frac{1}{1 - R_k^2} \quad (5)$$

where R_k is the negative correlation coefficient of the independent variable x_k for the regression analysis of the remaining independent variables. The larger the VIF, the greater the possibility of collinearity among independent variables. Therefore, it is critical to guarantee that variables with high VIF (VIF >5) are eliminated to ensure that the variables are independent of each other in the final model [32].

2.2.2. Backpropagation neural network (BPNN)

As shown in Fig. 1a, the BPNN is a neural network with a large number of neurons. All neurons in each layer are directly connected to the neurons in the next layer; hence, the BPNN can also be called a fully connected neural network. The BPNN contains an input layer, output layer, and series of intermediate or hidden layers. Each layer of neurons contains one or more neurons. The weights and biases of the BPNN are updated according to the gradient drop during training. Each part of BPNN is divided into several connection neuron layers [33]. The value of each neuron is

$$Y = f\left(\sum_{i=1}^{n} X_i * W_{ij} + b_j\right) \quad (6)$$

where X_i is the input variable, n is the number of neurons in the current layer, W_{ij} is the weight of the connection between the neuron and the next layer of neurons, b_j is the bias of the neuron, * represents the scalar product of two vectors, and f is the activation function. The neurons in the previous layer are all connected to each neuron in the current layer. A sigmoid function is a commonly used activation function that has an output value between 0 and 1. The specific formula is as follows:

$$\mathrm{sigmod}(x) = \frac{1}{1 + \mathrm{e}^{-x}}. \quad (7)$$

Backpropagation is a widely used training algorithm. Simultaneously, the BPNN is the most basic neural network model. Its output is propagated forward and the error is propagated backward. With the help of the returned error, the weights and biases can be updated, which finally achieves the purpose of optimizing the model. For the backpropagation of errors, the gradient descent method is generally used to update the weights. The first-order and second-order partial derivatives of all function variables of the error function are computed to obtain the gradient descent direction and speed of the function to determine the fastest descent direction, and correct the weights and thresholds of the network.

2.2.3. Genetic algorithm-backpropagation neural network (GA-BPNN)

In this study, we adopt a GA as an optimization method to adjust the weights and biases of the initial BPNN. A GA is the process of imitating biological evolution to select the most suitable results among all possible solutions. The optimization process mainly includes obtaining a large amount through selection, crossover, and mutation, in addition to selecting individuals with the best fitness, which is shown in Fig. 1b.

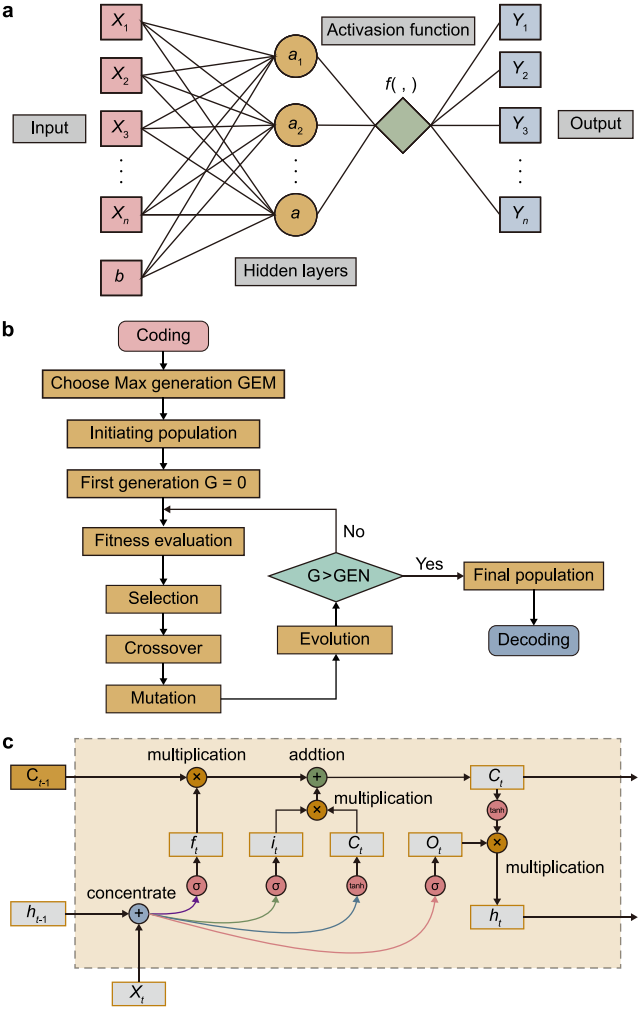

Fig. 1. Structure of the deep learning neural network model. **a**, Back Propagation Neural Network (BPNN). **b**, Genetic Algorithm (GA). **c**, Long Short Term Memory (LSTM) network.

Selection: The selection process is based on the fitness evaluation of individuals in the group: the fitter the individuals, the more offspring they produce, as shown in Equation (8).

Crossover: Crossover is the process of recombining two separate chromosomes to create a new individual. The calculation process is shown in Equation (9).

Mutation: The mutation operation randomly changes some of the values on the chromosome to create new individuals. Its calculation is shown in Equations (10) and (11):

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \quad (8)$$

$$a_{ij} = \begin{cases} a_{kj}(1-b) + a_{ij}b \\ a_{ij}(1-b) + a_{kj}b \end{cases} \quad (9)$$

$$a_{ij} = \begin{cases} a_{ij} + (a_{ij} - a_{max})*f(g) & r > 0.5 \\ a_{ij} + (a_{min} - a_{ij})*f(g) & r \le 0.5 \end{cases} \quad (10)$$

$$f(g) = r^2 * \left(1 - \frac{G}{G_{\max}}\right)^2 \quad (11)$$

where P_i is the selection probability of individual i, f_i is the fitness of individual i, and n is the number of individuals in the population. a_{ij} is the jth gene of the ith individual, a_{kj} is the jth gene of the kth individual, and $a_{\min}$ and $a_{\max}$ are the upper and lower bounds of the gene, respectively. G is the current iteration number, $G_{\max}$ is the maximum generation number, and r is a random number in the interval [0,1].

The optimization process consists of encoding and decoding the input, creating the initial population, calculating fitness, iterative operations, and adjusting the parameters. After the first generation is obtained, the most suitable individuals are selected from each generation according to the fitness result, and then a new generation is obtained using iterative operations until the set number of generations is reached. Therefore, the GA-BPNN is a method that first uses a GA to optimize the weights and biases that need to be set in advance for the BPNN, and then uses the most suitable coefficients set in advance to complete the training and testing of the BPNN.

2.2.4. Long short-term memory (LSTM)

The data flow of LSTM is similar to that of other recurrent neural networks (RNN) in that the data flow passes through each neuron using backforward propagation during training. The structural difference between LSTM and other RNNs is the difference in the results and functions of its neurons, which makes it an excellent solution to the problems of vanishing and exploding gradients [34], as shown in Fig. 1c.

The core aspects of the LSTM neural network are its storage cell form and gate structure. The memory cell is a way of disseminating previous data and can be considered as the memory of the network. The gate structure can be roughly divided into three types of gates: input gates, output gates, and forget gates. Each of these gates and memory cells are described in detail as follows:

Input Gate (I): The information input from the input layer at each moment first passes through the input gate, and the switch of the input gate determines whether the information is input into the memory cell at this moment, as shown in Equation (12).

Output Gate (O): The information output from the memory cell at each moment is determined by this gate, and its calculation is shown in Equations (13) and (14).

Forget Gate (F): Every time the value in the memory cell will undergo a process of choosing whether to be forgotten or not by the gate. If the data are marked, the value in the memory cell is cleared, that is, forgotten. The calculation process is shown in Equation (15).

Memory Cell (M): The information in the memory cell depends on the input at the previous moment and the forget gate. Additionally, at this moment, the information is input into the training process through the output gate. Its calculation is shown in Equation (16):

$$I_t = f(X_t W_{\mathrm{i}} + H_{t-1} W_{ih} + M_{t-1} W_{im} + b_{\mathrm{i}}) \quad (12)$$

$$O_t = f(X_t W_{\mathrm{o}} + H_{t-1} W_{oh} + M_{t-1} W_{om} + b_{\mathrm{o}}) \quad (13)$$

$$H_t = O_t * \tanh(M_{t-1}) \quad (14)$$

$$F_t = f\left(X_t W_{\mathrm{f}} + H_{t-1} W_{fh} + M_{t-1} W_{fm} + b_{\mathrm{f}}\right) \quad (15)$$

$$M_t = F_t * M_{t-1} + I_t * \tanh(X_t W_{\mathrm{m}} + H_{t-1} W_{mh} + b_{\mathrm{m}}), \quad (16)$$

where X_t represents the input variables; f is the activation function – in this model, we choose the sigmoid function (as shown in Equation (7)); W_{f}, W_{i}, W_{m}, and W_{o} are the weights of X_t in the forget gate, input gate, memory cell state, and output gate, respectively; W_{fh}, W_{ih}, W_{mh}, and W_{oh} are the weights of H_{t-1} at the forget gate, input gate, memory cell state, and output gate, respectively; W_{fm}, W_{im}, and W_{om} are weights related to the connection between the memory cell state and different structures; b_{f}, b_{i}, b_{c}, and b_{o} are the biases in the each structure, respectively; and * represents the scalar product of two vectors. (The other variables not given were defined in previous equations.)

The backpropagation algorithm is used throughout the training process of the LSTM, and the associated variable matrix is continuously optimized to finally determine the optimal set of variables. The problems of exploding and vanishing gradients during training and learning are easily solved by LSTM [35].

2.3. Model performance evaluation

In this study, we use two performance evaluation metrics: relative root-mean-square error (RMSE) and coefficient of determination (R^2). The RMSE measures the deviation between observations and true values; the formula is shown in Equation (17). R^2 is generally used in regression models to evaluate the conformity between the predicted and actual values, which is calculated as shown in Equation (18):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(y_t^{\mathrm{act}} - y_t^{\mathrm{pre}}\right)^2} \quad (17)$$

where $\bar{y}_t^{\mathrm{act}}$ is the actual value and y_t^{pre} is the predictive value; and

$$R^2 = 1 - \frac{\sum_{t=1}^{n}\left(y_t^{\mathrm{act}} - y_t^{\mathrm{pre}}\right)^2}{\sum_{t=1}^{n}\left(y_t^{\mathrm{act}} - \bar{y}_t^{\mathrm{act}}\right)^2} \quad (18)$$

where y_t^{act} represents the actual value, y_t^{pre} represents the predictive value, and $\bar{y}_t^{\mathrm{act}}$ represents the average of the actual data values.

2.4. Description of the experimental data

The set of plant data used in this study originated from a CW located in a city in southern China, with a total construction area of 42,500 m^2 (31,000 m^2 is a vertical flow CW) (as shown in Fig. 2a). It undertakes 20,000 m^2 tail water from the first phase of the upstream Longhua Wastewater Treatment Plant every day.

We took sampling points at 10:00 in the morning every day. The dataset included the following environmental indicators: meteorological indicators (temperature, relative humidity, and rainfall), water quantity indicators (flow velocity), water quality indicators (NH_4^+-N'$_{inf}$, TP_{inf}, COD_{inf}, SS_{inf}, PH, $BOD_{5\text{-}inf}$, NH_4^+-N'$_{eff}$, TP_{eff}, and COD_{eff}). We surveyed and collected meteorological indicators at sampling points from the local meteorological bureau, whereas water quality indicators and water quantity indicators sampled and collected from sampling points. The cumulative number of days for data collection was 186 days (from January 28, 2021 to August 31, 2021) However, some raw data exhibited diverse and irregular patterns, which implied that data-driven modeling would fail to

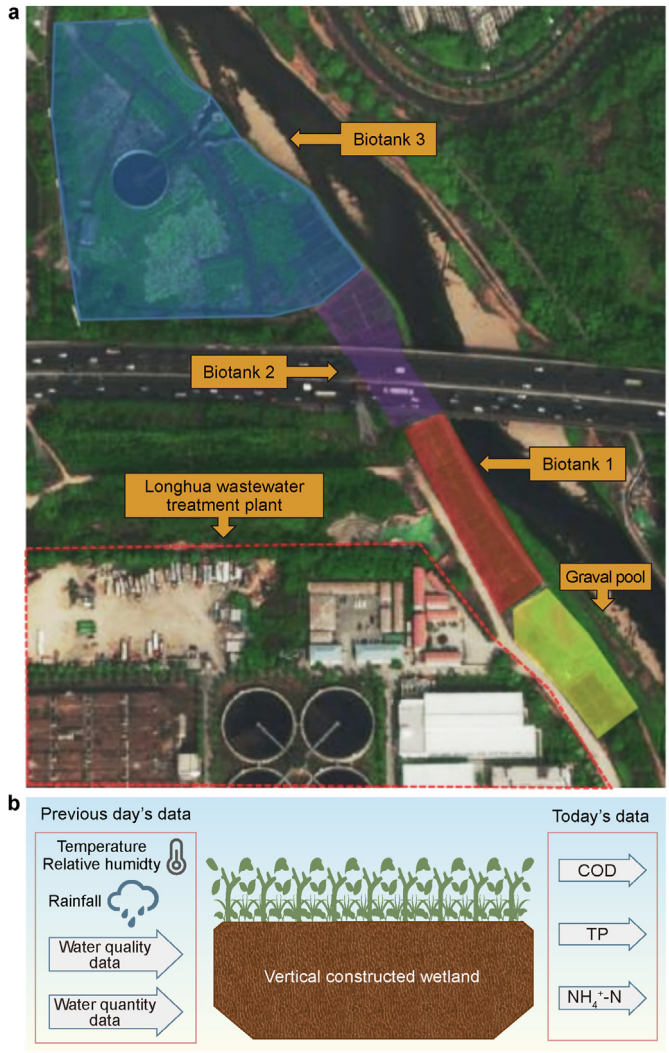

Fig. 2. Diagram and model description of the constructed wetland. **a**, Satellite photo. **b**, Prediction model.

achieve great model performance. The structure of our model is shown in Fig. 2b.

We performed moving average processing on each water effluent indicator using Equation (1). Therefore, three moving average indicators plus 13 environmental indicators provided a total of 16 indicators, a total of 2960 indicators. Table 1 illustrates the average, standard deviation, minimum, and maximum values of the 16 indicators.

We divided the dataset into two subsets, that is, the training set (January 29, 2021 to July 13, 2021, 166 days) and testing set (July 14, 2021 to August 31, 2021, 19 days), which corresponded to a total of 90% and 10% data for training and testing, respectively. We mainly used the training set to train the parameters in the neural network, which is associated with input-output models. We used the testing set to verify the performance of the model. After training on the training set, we compared and assessed the performance of each model using the testing set.

2.5. Computing environment

We implemented the MLR model using SPSS 22.0 software. We implemented the BPNN, GA-BPNN, and LSTM models in MATLAB 2020b using the Neural Network Toolbox, Genetic Algorithm Optimization Toolbox, and Deep Learning Toolbox.

3. Results and discussion

3.1. Raw data analysis

Through continuous monitoring of the influent and effluent, we analyzed the basic variation rules of water quality in the large-scale CW. Fig. 3 shows the concentration of TP, COD, and NH_4^+-N, and the removal efficiency for each indicator (Text S1). It is obvious that, in most cases, CWs had a certain removal effect on pollutants; however, there were still cases in which there was no removal effect. There may be three main reasons for these results: (1) The concentration of pollutants in the influent water was too low, which led to the description of the substances in the original soil of the wetland and induced the increase of pollutant concentration in the wetland. For example, the concentration of TP and NH_4^+-N in the water was too low, which resulted in the low removal rate of wetlands on the 145th to 180th days. (2) The COD:TP ratio in the tail water of the sewage treatment plant was too low. For example, the COD:TP ratio was significantly lower than 100:1 around the 5th and 40th days, which resulted in an insufficient carbon source, which was not conducive to the removal of phosphorus in water. (3) The pollutant removal efficiencies of CWs are greatly affected by external conditions, such as temperature and rainfall. During strong rainfall, the concentrations of pollutants in water are affected. For these reasons, the effluent quality of the CW in the actual

Table 1
Summary statistics for the 16 variables.

variables	Indicators	Max value	Min value	Average value	Standard deviation
v_1	temperature	32.8	14.4	25.99	4.72
v_2	Relative humidity	100	30	64.58	11.98
v_3	rainfall	84.8	0	4.0135	11.82
v_4	flow velocity	35,442	10,576	17,372.816	4211.12
v_5	$NH_4^+-N'_{inf}$	0.9	0.009	0.1585	0.1456
v_6	TP_{inf}	0.83	0.004	0.09427	0.073
v_7	COD_{inf}	25.31	0.053	15.14	3.277
v_8	SS_{inf}	7	1	3.357	0.88
v_9	PH	7.98	5.61	7.21	0.327
v_{10}	$BOD_{5\text{-}inf}$	5.6	0.8	3.0196	0.637
v_{11}	$NH_4^+-N'_{eff}$	0.546	0.006	0.121	0.104
v_{12}	TP_{eff}	0.325	0.012	0.0889	0.039
v_{13}	COD_{eff}	22	0.076	13.935	3.14
v_{14}	$NH_4^+-N'_{eff(ma)}$	0.351	0.033	0,1211	0.0694
v_{15}	$TP_{eff(ma)}$	0.213	0.0253	0.089	0.0257
v_{16}	$COD_{eff(ma)}$	17.393	6.58	13.903	1.943

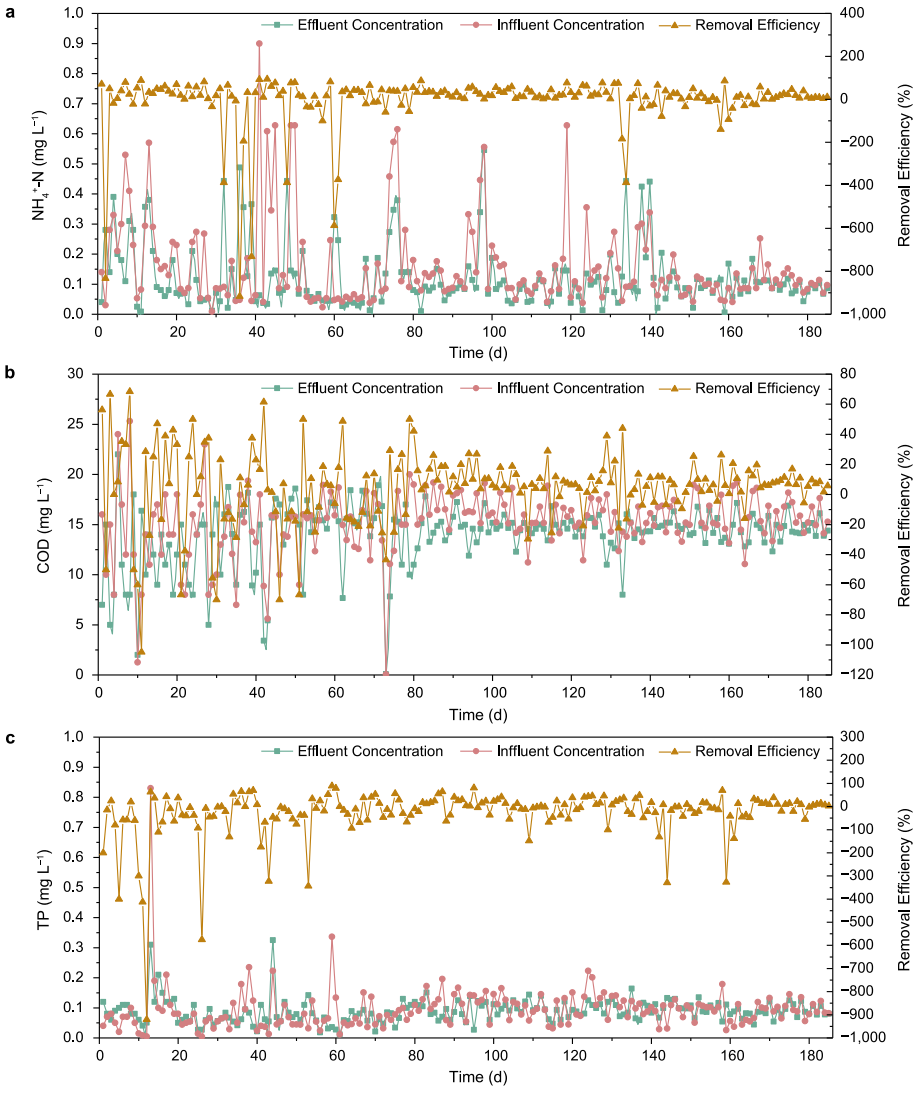

Fig. 3. Water quality parameters measured from influent and effluent in the study constructed wetland. **a**, NH_4^+-N. **b**, COD. **c**, TP.

environment was similar to that in the specific environment in the laboratory; that is, it was generally still lower than the discharge standard. However, the complexity of the data and model characteristics in the actual environment was much higher than that in the laboratory during the construction of the data-driven model.

3.2. Structure determination and model results

3.2.1. MLR modeling result

For MLR models, it is necessary to ensure that the variables are independent of each other and not affected by multicollinearity problems. Fortunately, the VIF values of the ten independent variables in the MLR model were all small, such as NH_4^+-N $_{inf}$ and PH being 1.11 and 1.07. The remaining VIF values were between 1.18 and 2.119, that is, all less than 5. This demonstrates that the correlation between independent variables was small and there was no multicollinearity problem. All the results are shown in Table 2. Therefore, we used the two subsets described in Section 3.2 to train and test the model, and calculated the regression coefficient of the model using regression analysis. The detailed results of MLR modeling are shown in Table 3.

3.2.2. Neural network modeling results

The neural network models were used by the two back-propagation algorithms (BPNN and LSTM) during the entire training process. Additionally, we used a GA to optimize the weights and biases of the BPNN as the third network model. To the best of our knowledge, the structure of a network model is determined by the quantity of layers, total number of neurons in each layer, and characteristic of the transmission functions, and is a vital

Table 2
Multicollinearity analysis results of independent variables in MLR model.

Input indicator	Temp	RH	Rainfall	Flow	NH_4^+-N $_{inf}$	TP_{inf}	COD_{inf}	SS_{inf}	PH_{inf}	$BOD_{5\text{-}inf}$
Independent	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
Variable VIF	1.18	1.30	1.34	1.17	1.11	1.25	2.12	1.17	1.07	2.09

Table 3
The MLR model equations.

Output indicator	Response variable	Model equation
NH_4^+-N (ma)	$Y_{NH4\text{-}N'eff\ (ma)}^+$	$-0.002x_1+0.001x_2-0.001x_3-1.904\times10^{-6}x_4+0.184x_5+0.001x_6-0.003x_7+0.007x_8+0.018x_9+0.025x_{10}-0.065$
NH_4^+ −N	$Y_{NH4\ -N'eff}^+$	$-0.002x_1+0.001x_2-0.001x_3-9.459\times10^{-7}x_4+0.253x_5+0.164+x_6$-$0.001x_7+0.009x_8+0.052x_9+0.008x_{10}-0.346$
COD (ma)	$Y_{CODeff\ (ma)}$	$-0.079x_1+0.038x_2+0.011x_3+4.994\times10^{-5}x_4-1.479x_5+0.734x_6+0.448x_7-0.077x_8+1.337x_9-0.697x_{10}-5.918$
COD	Y_{CODeff}	$0.139x_1+0.036x_2+0.003x_3+4.222\times10^{-5}x_4-3.669x_5+1.933x_6+0.119x_7+0.046x_8+0.214x_9+0.086x_{10}+3.939$
TP (ma)	$Y_{TPeff\ (ma)}$	$0.000314x_1-0.001x_2+0.00046x_3+5.143\times10^{-7}x_4+0.009x_5+0.108+x_6+0.001x_7-0.000359x_8-0.002x_9-0.001x_{10}+0.112$
TP	Y_{TPeff}	$0.000348x_1-0.000482x_2+0.001x_3+6.324\times10^{-7}x_4+0.025x_5+0.274_+x_6-1.138\times10^{-5}x_7+0.001x_8-0.001x_9+0.001x_{10}+0.069$

part of model development. Increasing the number of neurons could improve the accuracy of nonlinear fitting. However, an overly complex network would lead to overfitting and prolong the training time. Therefore, all applied models in this study had an input layer with ten neurons, corresponding to Temp, RH, Rainfall, Flow, NH_4^+-N $_{inf}$, TP_{inf}, COD_{inf}, SS_{inf}, PH_{inf}, and $BOD_{5\text{-}inf}$. The output layer was composed of six neurons, corresponding to effluent concentrations of NH_4^+-N $_{eff}$, TP_{eff}, COD_{eff}, NH_4^+-$N_{eff(ma)}$, $TP_{eff\ (ma)}$, and $COD_{eff\ (ma)}$. Additionally, for the three models, we conducted experiments on one to four hidden layer structures, where we attempted to use 3−30 neurons in each hidden layer.

Considering the training efficiency and prediction accuracy, the resulting optimal topology of the hidden layers for the BPNN model was a three-layer structure, with 18 neurons in hidden layer 1, 14 neurons in hidden layer 2, and six neurons in hidden layer 3 (Fig. S1). Additionally, the best performing GA-BPNN had three hidden layers, with 16 neurons in layer 1, 11 neurons in layer 2, and 8 neurons in layer 3 (Figs. S2 and S3). The optimal structure of LSTM had three hidden layers, with 17 neurons in layer 1, 14 neurons in layer 2, and 12 neurons in layer 3 (Figs. S4 and S5).

3.3. Prediction performance on the raw testing set

A comparison of predicted versus measured data for three water quality indicators (COD_{eff}, NH_4^+-N $_{eff}$, and TP_{eff}) is shown in Fig. 4. Different types of models had very different prediction results. The MLR predictions had a high degree of oscillation, and their R^2 values were all less than 0.32 (as shown in Fig. 4). Even when NH_4^+-N $_{eff}$ was predicted (as shown in Fig. 4a), it was only 0.225, which means that the prediction of the effluent quality of CWs is not a simple linear problem. In comparison, the prediction results of the BPNN were much better, and its R^2 was greater than 0.7; however, this is still far from satisfactory. In predicting COD_{eff} (as shown in Fig. 4b), the BPNN underestimated the peak COD_{eff} concentration, which resulted in a smooth line. The inconsistency of the BPNN suggests that it performed poorly compared with LSTM. However, when we added a GA to optimize the BPNN, although the GA-BPNN was unable to match the accuracy of LSTM, the GA-BPNN still achieved an R^2 of 0.81, which was higher than that of a single BPNN. As shown in Fig. 4, the prediction effect of using the weights and bias generated by the GA to reduce the RMSE was much higher than that of the neural network generated by randomly generated weights and biases. LSTM outperformed the other models in the prediction of all metrics, particularly in the prediction of COD_{eff} (as shown in Fig. 4b), where LSTM substantially outperformed the other models, with an R^2 of 0.93. The reason for the satisfactory performance of LSTM may be that it can take into account the influence of past results on the present, which plays an important role in time series problems.

3.4. Effect of the moving average on prediction performance

A comparison of predicted versus measured data for three water quality indicators after the moving average ($COD_{eff(ma)}$, NH_4^+-N $_{eff(ma)}$, and $TP_{eff(ma)}$) is shown in Fig. 5. After we used the moving average method, the processed data were much smoother than the original data. We recreated new models using the processed data, and the accuracy of each model improved considerably. The improvement of the GA-BPNN when we used the moving average method was the most substantial among the four models, and the R^2 of the three types of water quality indicators was close to 0.9, or even higher. By contrast, the accuracy of LSTM also improved; however, the increased amplitude was not as obvious as for the other models. Only in the prediction of NH_4^+-N $_{eff(ma)}$ did R^2 achieve an increase of 0.013 compared with the original data (as shown in Figs. 4a and 5a). We speculate that the application of the moving average method enabled the other three models, except LSTM, to consider the influence of past results so that high-frequency errors were eliminated, thereby improving accuracy.

3.5. Comparison of the models

By comparing RMSE and R^2 (as shown in Fig. 6), we can more intuitively identify the strength of the predictability of the four models. For the original dataset, based on the MLR model, the RMSE of the BP model decreased considerably, and R^2 for the COD_{eff}, TP_{eff}, and NH_4^+-N $_{eff}$ prediction results increased by 49.1%, 47.2%, and 43.2%, respectively. This suggests that traditional machine learning can solve multiple regression problems better than linear methods because machine learning can fit more complex functions and achieve higher accuracy. However, because of the influence of the possible local minimum problem, the accuracy of the prediction results obtained by the BPNN only was still not satisfactory. After we optimized the BPNN using a GA, the RMSE of each model further decreased, and the R^2 of the three predictors increased by 8.55%, 6.4%, and 7.31%, respectively. The reason for this is that we optimized the weights and biases of the network with the goal of reducing the RMSE of the prediction results. After we compared LSTM with the GA-BPNN, the RMSE of LSTM decreased more substantially, and R^2 for each indicator increased by 9.9%, 10.49%, and 7.8% sequentially. This is because water quality data are complex time series data, and LSTM considers the effect of past results on the present, thereby achieving higher prediction accuracy.

Finally, after we processed the original data using the moving average method, the accuracy of the results of each model improved because some noise was removed. The improvement effect on the GA-BPNN was the most notable, and the increase in R^2 reached above 8%, on average, whereas the R^2 of LSTM was only 2%. We assume that this is because we averaged three days of data in the smoothing process, which transferred the previous influence into the other models; however, LSTM considered the influence of previous data, and thus achieved an insignificant improvement.

3.6. Future perspectives

In the future, we will attempt to develop a hybrid algorithm of RNNs to achieve higher accuracy or a faster model construction speed. Additionally, the prediction effect of the neural network had

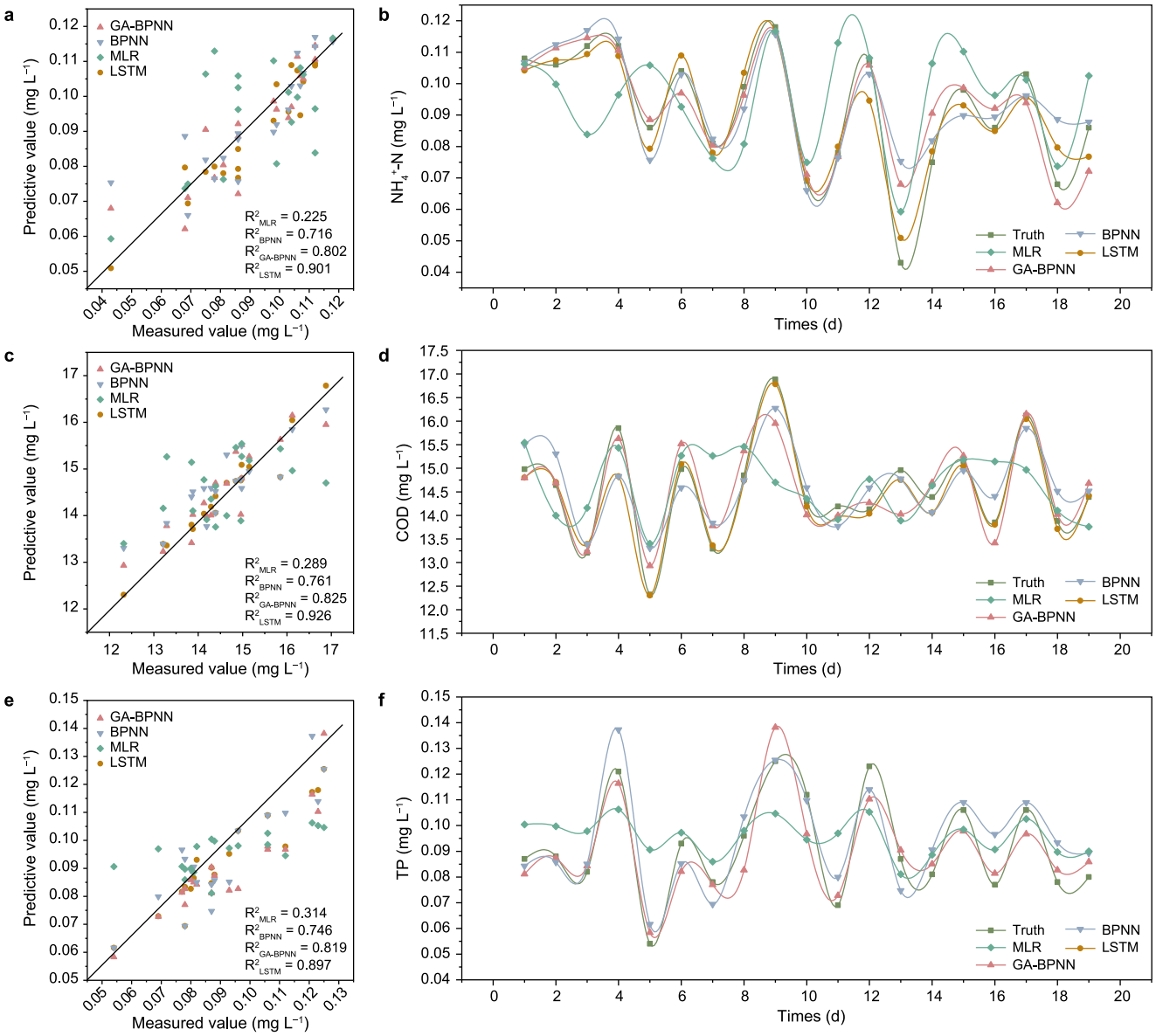

Fig. 4. Comparison of the three water quality indices predicted by the MLR model, the BPNN model, the GA-BPNN model, and the LSTM model with the measured results and their corresponding R^2 values. **a**–**b**, Scatter plot (**a**) and line plot (**b**) for NH_4^+-N_{eff}. **c**–**d**, Scatter plot (**c**) and line plot (**d**) for COD_{eff}. **e**–**f**, Scatter plot (**e**) and line plot (**f**) for TP_{eff}.

a high correlation with the amount of input data; however, an excessively high amount of data leads to a large consumption of human and material resources. Therefore, on the premise of not affecting the prediction effect of the model, we will also attempt to reduce the amount of data used. Additionally, we will further improve the forecast model of CWs to analyze GHG emissions. The timely prediction of carbon emissions or the absorption of CWs is important for helping the entire urban system to achieve carbon neutrality and further improve the intelligent management of urban water environments.

4. Conclusion

The deep learning network successfully predicted the next-day effluent quality of large-scale CWs and reveals the mapping relationship between the collected multi-source datasets and effluent quality. By comparing the prediction effects of the four models for three water effluent indicators, we obtained three main research conclusions: (1) Based on the original data with large fluctuations, the moving average method can be used to remove high-frequency noise in an actual large-scale application, and smoothed data can be obtained to improve the prediction effect. (2) Compared with MLR, backward feedback neural network, and neural network based on GA optimization, a deep learning neural network (LSTM) that can take into account previous training results achieves a better prediction effect on time series problems, such as water quality prediction. (3) A deep learning network can be quickly established to predict water quality in a real scenario by collecting a large number of simple and easy-to-obtain water quality indicators. The LSTM neural network can solve the disadvantage of time and money wasting to perform miniature experiments to obtain various parameters in the modeling of CWs. With the widespread application of CW sewage treatment methods, the prediction of CWs' effluent quality not only plays a crucial role in the regulation of the urban

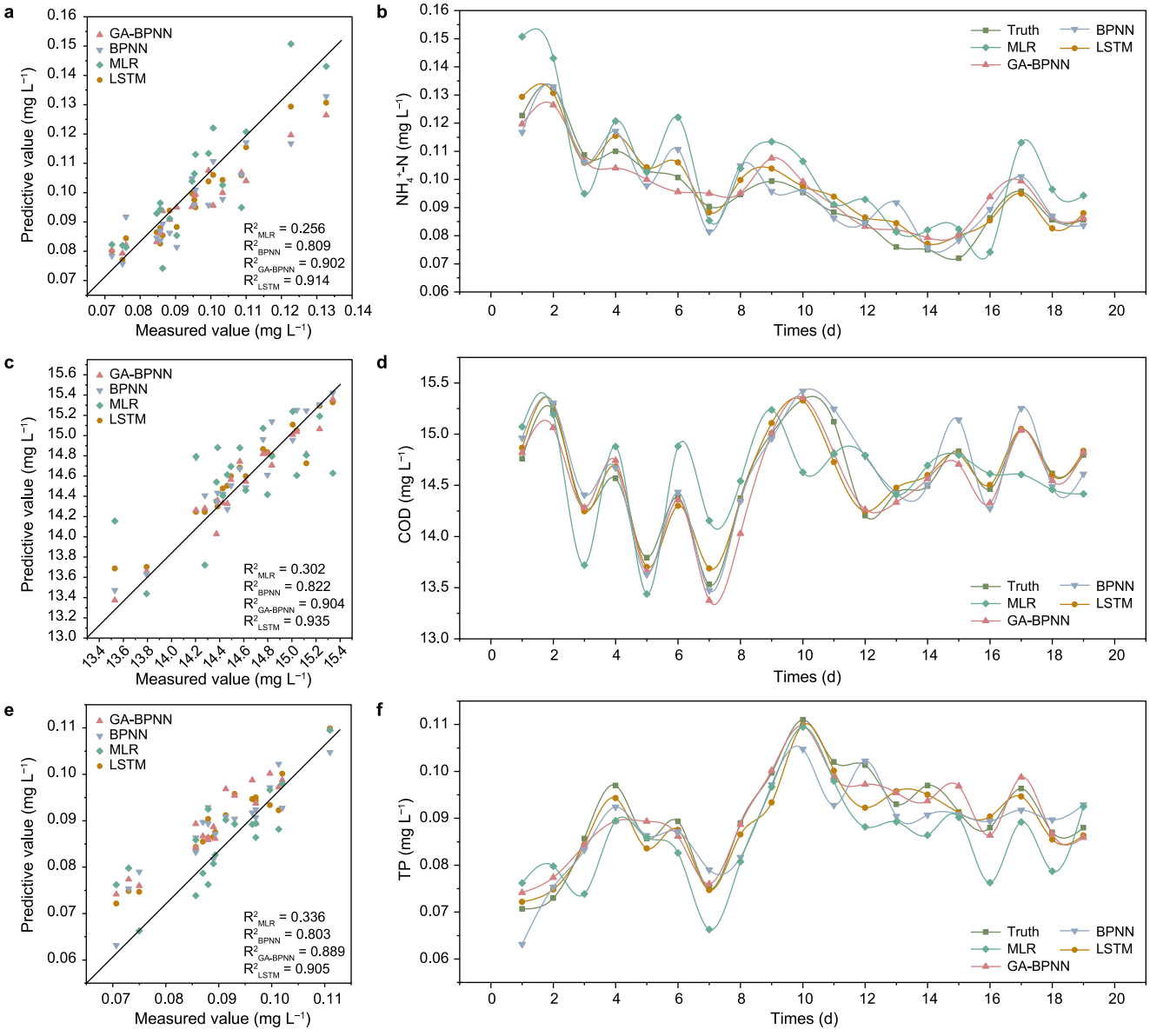

Fig. 5. Comparison of the three water quality indices after the moving average predicted by the MLR model, the BPNN model, the GA-BPNN model, and the LSTM model with the measured results and their corresponding R^2 values. **a**–**b**, Scatter plot (**a**) and line plot (**b**) for NH_4^+-$N_{eff(ma)}$. **c**–**d**, Scatter plot (**c**) and line plot (**d**) for $COD_{eff(ma)}$. **e**–**f**, Scatter plot (**e**) and line plot (**f**) for $TP_{eff(ma)}$.

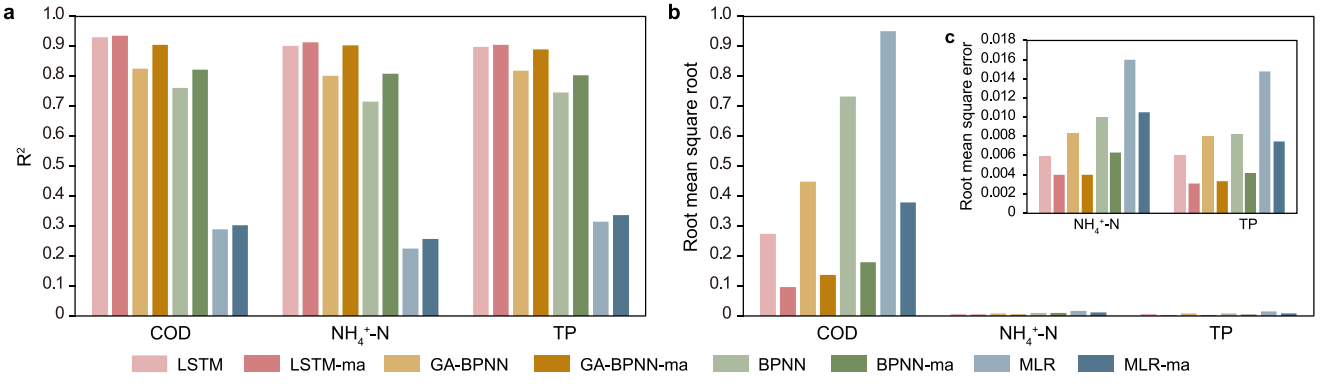

Fig. 6. Accuracy evaluations for MLR, BPNN, GA-BPNN, and LSTM models. **a**, R^2 comparison. **b**, RMSE comparison. **c**, RMSE comparison for NH_4^+-N and TP with more details.

water environment but also provides a feasible basis for solving urban non-point source pollution.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This investigation was funded by National Natural Science Foundation of China (No. 51908161 & 52100044), Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515010807), State Key Laboratory of Urban Water Resource and Environment (Harbin Institute of Technology) (2021TS30) and Shenzhen Science and Technology Program (No. KQTD20190929172630447, KCXFZ20211020163404007 and GXWD20201230155427003-20200824100026001).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ese.2022.100207.

References

[1] J.L. Han, Z.N. Yang, H. Wang, H.Y. Zhong, D. Xu, S. Yu, L. Gao, Decomposition of pollutants from domestic sewage with the combination systems of hydrolytic acidification coupling with constructed wetland microbial fuel cell, J. Clean. Prod. 319 (2021), https://doi.org/10.1016/j.jclepro.2021.128650.

[2] D. Li, Z. Chu, M. Huang, B. Zheng, Multiphasic assessment of effects of design configuration on nutrient removal in storing multiple-pond constructed wetlands, Bioresour. Technol. 290 (2019), https://doi.org/10.1016/j.biortech.2019.121748.

[3] A. Delre, M. ten Hoeve, C. Scheutz, Site-specific carbon footprints of Scandinavian wastewater treatment plants, using the life cycle assessment approach, J. Clean. Prod. 211 (2019) 1001–1014, https://doi.org/10.1016/j.jclepro.2018.11.200.

[4] H.-T. Shi, X.-C. Feng, Z.-J. Xiao, W.-Q. Wang, Y.-M. Wang, X. Zhang, Y.-J. Xu, N.-Q. Ren, Analysis of the β-cyclodextrin enhancing bio-denitrification from the perspective of substrate metabolism, electron transfer, and iron acquisition, Chem. Eng. J. 446 (2022), 137358, https://doi.org/10.1016/j.cej.2022.137358.

[5] Y. Liang, H. Zhu, G. Banuelos, B. Yan, B. Shutes, X. Cheng, X. Chen, Removal of nutrients in saline wastewater using constructed wetlands: plant species, influent loads and salinity levels as influencing factors, Chemosphere 187 (2017) 52–61, https://doi.org/10.1016/j.chemosphere.2017.08.087.

[6] W.S. Birch, M. Drescher, J. Pittman, R.C. Rooney, Trends and predictors of wetland conversion in urbanizing environments, J. Environ. Manag. 310 (2022) 114723, https://doi.org/10.1016/j.jenvman.2022.114723.

[7] J. Persson, H.B. Wittgren, How hydrological and hydraulic conditions affect performance of ponds, Ecol. Eng. 21 (4-5) (2003) 259–269, https://doi.org/10.1016/j.ecoleng.2003.12.004.

[8] T.-M. Su, S.-C. Yang, S.-S. Shih, H.-Y. Lee, Optimal design for hydraulic efficiency performance of free-water-surface constructed wetlands, Ecol. Eng. 35 (8) (2009) 1200–1207, https://doi.org/10.1016/j.ecoleng.2009.03.024.

[9] X. Wang, F. Zhang, J. Ding, H.-t. Kung, A. Latif, V.C. Johnson, Estimation of soil salt content (SSC) in the Ebinur Lake Wetland National Nature Reserve (ELWNNR), Northwest China, based on a Bootstrap-BP neural network model and optimal spectral indices, Sci. Total Environ. 615 (2018) 918–930, https://doi.org/10.1016/j.scitotenv.2017.10.025.

[10] H. Wang, D. Xu, J. Han, R. Xu, D. Han, Reshaped structure of microbial community within a subsurface flow constructed wetland response to the increased water temperature: improving low-temperature performance by coupling of water-source heat pump, Sci. Total Environ. 781 (2021), https://doi.org/10.1016/j.scitotenv.2021.146798.

[11] J. Zhang, H. Sun, W. Wang, Z. Hu, X. Yin, N. Huu Hao, W. Guo, J. Fan, Enhancement of surface flow constructed wetlands performance at low temperature through seasonal plant collocation, Bioresour. Technol. 224 (2017) 222–228, https://doi.org/10.1016/j.biortech.2016.11.006.

[12] A.N. Ahmed, F.B. Othman, H.A. Afan, R.K. Ibrahim, C.M. Fai, M.S. Hossain, M. Ehteram, A. Elshafie, Machine learning methods for better water quality prediction, J. Hydrol. 578 (2019), https://doi.org/10.1016/j.jhydrol.2019.124084.

[13] M. Hameed, S.S. Sharqi, Z.M. Yaseen, H.A. Afan, A. Hussain, A. Elshafie, Application of artificial intelligence (AI) techniques in water quality index prediction: a case study in tropical region, Malaysia, Neural Comput. Appl. 28 (2017) S893–S905, https://doi.org/10.1007/s00521-016-2404-7.

[14] R. Samso, J. Garcia, P. Molle, N. Forquet, Modelling bioclogging in variably saturated porous media and the interactions between surface/subsurface flows: application to Constructed Wetlands, J. Environ. Manag. 165 (2016) 271–279, https://doi.org/10.1016/j.jenvman.2015.09.045.

[15] N.-B. Chang, G. Mohiuddin, A.J. Crawford, K. Bai, K.-R. Jin, Diagnosis of the artificial intelligence-based predictions of flow regime in a constructed wetland for stormwater pollution control, Ecol. Inf. 28 (2015) 42–60, https://doi.org/10.1016/j.ecoinf.2015.05.001.

[16] F. Granata, R. Gargano, G. de Marinis, Artificial intelligence based approaches to evaluate actual evapotranspiration in wetlands, Sci. Total Environ. 703 (2020), https://doi.org/10.1016/j.scitotenv.2019.135653.

[17] A. Hosseinzadeh, M. Baziar, H. Alidadi, J.L. Zhou, A. Altaee, A.A. Najafpoor, S. Jafarpour, Application of artificial neural network and multiple linear regression in modeling nutrient recovery in vermicompost under different conditions, Bioresour. Technol. 303 (2020), https://doi.org/10.1016/j.biortech.2020.122926.

[18] B.P.L. Lau, S.H. Marakkalage, Y. Zhou, N. Ul Hassan, C. Yuen, M. Zhang, U.X. Tan, A survey of data fusion in smart city applications, Inf. Fusion 52 (2019) 357–374, https://doi.org/10.1016/j.inffus.2019.05.004.

[19] G. Niu, X. Li, X. Wan, X. He, Y. Zhao, X. Yi, C. Chen, L. Xujun, G. Ying, M. Huang, Dynamic optimization of wastewater treatment process based on novel multi-objective ant lion optimization and deep learning algorithm, J. Clean. Prod. 345 (2022), 131140, https://doi.org/10.1016/j.jclepro.2022.131140.

[20] K. Song, Y.-S. Park, F. Zheng, H. Kang, The application of Artificial Neural Network (ANN) model to the simulation of denitrification rates in mesocosm-scale wetlands, Ecol. Inf. 16 (2013) 10–16, https://doi.org/10.1016/j.ecoinf.2013.04.002.

[21] G. Niu, X. Yi, C. Chen, X. Li, D. Han, B. Yan, M. Huang, G. Ying, A novel effluent quality predicting model based on genetic-deep belief network algorithm for cleaner production in a full-scale paper-making wastewater treatment, J. Clean. Prod. 265 (2020), https://doi.org/10.1016/j.jclepro.2020.121787.

[22] R. Boutaba, M.A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, O.M. Caicedo, A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, Journal of Internet Services and Applications 9 (2018), https://doi.org/10.1186/s13174-018-0087-2.

[23] Z. Yu, K. Yang, Y. Luo, C. Shang, Spatial-temporal process simulation and prediction of chlorophyll-a concentration in Dianchi Lake based on wavelet analysis and long-short term memory network, J. Hydrol. 582 (2020), https://doi.org/10.1016/j.jhydrol.2019.124488.

[24] Y. Zhang, C. Li, Y. Jiang, L. Sun, R. Zhao, K. Yan, W. Wang, Accurate prediction of water quality in urban drainage network with integrated EMD-LSTM model, J. Clean. Prod. (2022), 131724, https://doi.org/10.1016/j.jclepro.2022.131724.

[25] 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, Chem. Eng. J. 143 (1-3) (2008) 96–110, https://doi.org/10.1016/j.cej.2007.12.029.

[26] C.S. Akratos, J.N.E. Papaspyros, V.A. Tsihrintzis, Artificial neural network use in ortho-phosphate and total phosphorus removal prediction in horizontal subsurface flow constructed wetlands, Biosyst. Eng. 102 (2) (2009a) 190–201, https://doi.org/10.1016/j.biosystemseng.2008.10.010.

[27] C.S. Akratos, J.N.E. Papaspyros, V.A. Tsihrintzis, Total nitrogen and ammonia removal prediction in horizontal subsurface flow constructed wetlands: use of artificial neural networks and development of a design equation, Bioresour. Technol. 100 (2) (2009b) 586–596, https://doi.org/10.1016/j.biortech.2008.06.071.

[28] P. Antwi, J.Z. Li, J. Meng, K.W. Deng, F.K. Quashie, J.L. Li, P.O. Boadi, Feedforward neural network model estimating pollutant removal process within mesophilic upflow anaerobic sludge blanket bioreactor treating industrial starch processing wastewater, Bioresour. Technol. 257 (2018) 102–112, https://doi.org/10.1016/j.biortech.2018.02.071.

[29] C. Kiiza, S.-q. Pan, B. Bockelmann-Evans, A. Babatunde, Predicting pollutant removal in constructed wetlands using artificial neural networks (ANNs), Water Sci. Eng. 13 (1) (2020) 14–23, https://doi.org/10.1016/j.wse.2020.03.005.

[30] S. Hwangbo, R. Al, X. Chen, G. Sin, Integrated model for understanding N2O emissions from wastewater treatment plants: a deep learning approach, Environ. Sci. Technol. 55 (3) (2021) 2143–2151, https://doi.org/10.1021/acs.est.0c05231.

[31] S.B. Vilsen, D.-I. Stroe, Battery state-of-health modelling by multiple linear regression, J. Clean. Prod. 290 (2021), https://doi.org/10.1016/j.jclepro.2020.125700.

[32] I.M. Herrig, S.I. Boeer, N. Brennholt, W. Manz, Development of multiple linear regression models as predictive tools for fecal indicator concentrations in a stretch of the lower Lahn River, Germany, Water Res. 85 (2015) 148–157, https://doi.org/10.1016/j.watres.2015.08.006.

[33] D. Gebler, G. Wiegleb, K. Szoszkiewicz, Integrating river hydromorphology and water quality into ecological status modelling by artificial neural networks, Water Res. 139 (2018) 395–405, https://doi.org/10.1016/j.watres.2018.04.016.
[34] J. Liu, Z. Wang, M. Xu, DeepMTT: a deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network, Inf. Fusion 53 (2020) 289–304, https://doi.org/10.1016/j.inffus.2019.06.012.
[35] D. Niu, F. Wu, S. Dai, S. He, B. Wu, Detection of long-term effect in forecasting municipal solid waste using a long short-term memory neural network, J. Clean. Prod. 290 (2021), https://doi.org/10.1016/j.jclepro.2020.125187.