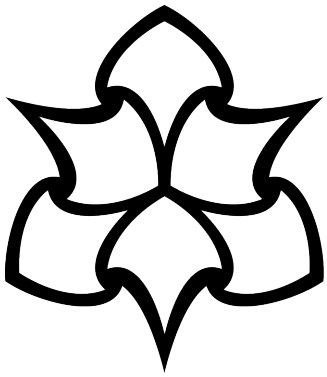

# Graph embedding-based intelligent industrial decision for complex sewage treatment processes

**Zhiwei Guo**[1] **Yu Shen**[1] **Ali Kashif Bashir**[2]
**Keping Yu**[3] **Jerry Chun-wei Lin**[4]

[1]School of Artificial Intelligence, National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing, China

[2]Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK

[3]Global Information and Telecommunication Institute, Waseda University, Shinjuku, Tokyo, Japan

[4]Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway

**Abstract**

Intelligent algorithms-driven industrial decision sys-tems have been a general demand for modeling complex sewage treatment processes (STP). Existing researches modeled complex STP with the use of var-ious neural network models, yet neglecting the fact that latent and occasional relations exist inside complex STP. To deal with the challenge, this paper proposes graph embedding-based intelligent industrial decision for complex STP (GE-STP). The graph embedding (GE) scheme is employed to enhance feature extraction and neural computing structure is utilized to simulate uncertain biochemical transformation inside STP. The introduction of GE can not only improves the fineness of feature spaces, but also improves the representative ability of models towards complex industrial processes. On this basis, the GE-STP is evaluated on a real-world data set collected from a realistic sewage treatment plant equipped with a set of Internet of Things devices. And some typical neural network models that have been utilized for modeling complex STP, are selected as baseline methods. Three groups of experiments show that efficiency of the GE-STP exceeds baselines about 6%–12%, and that the GE-STP is not susceptible to parameter changing.

## INTRODUCTION

In contemporary society, water resource has become a kind of important energy related to sustainable development.[1] In this context, it is of great importance to realize optimal management of water resource. One of the core tasks lies in the monitoring and prediction towards the operation quality of sewage treatment process (STP). STP is a typical industrial process driven by biochemical reaction, accompanied by invisible material exchange and energy transfer. Therefore, it is usually filled with uncertainty and complexity, which makes the modeling of it quite challenging. To predict the results of STP more accurately, it is expected to establish an effective process model that expresses STP.[2] The most intuitive strategy is to establish a process model based on biochemical reaction knowledge. But due to a large number of redundant process parameters, such type of biochemical mechanism-driven methods often faced the problem of low computational efficiency in practice.[3] Alternatively, novel insights can be provided by cross-domain technologies,[4] such as the Internet of Things (IoT) that possesses the strong ability of data collection and management.[5] On this basis, the data analysis algorithm is embedded into the interface, forming data-driven modeling schemes for general industrial processes.[6] It abstracts STP from the perspective of statistics, rather than the utilization of biochemical mechanisms. With a large number of data used for model training, data-driven models can be formulated to approximately express STP. Therefore, intelligent computing seems to be a promising modeling scheme for complex STP.

During the past few years, intelligent computing-based modeling strategies for STP had been widely noticed by relevant researchers, yielding a huge number of typical technical approaches.[7-21] To sum up, almost all of them originated from neural networks, and were exactly different modified versions of neural networks. For instance, Hassen and Asmare[22] developed a neural network with both forward and backward propagation directions to predict outlet results of STP. Ruan et al.[23] introduced fuzzy logic to enhance inference ability by putting forward a fuzzy neural network model. Huang et al.[9] exploited wavelet operation to accelerate calculation speed via the proposal of a fuzzy wavelet neural network. Nevertheless, existing methods still suffer from some drawbacks or limitations when it comes to solution thoughts. In real-world scenarios of sewage treatment plants, treatment pools are usually not an integrated one. To improve processing efficiency, instead, nearly all the plants separate their pools into a number of branches parallel subpools. Taking a typical plant located in Chongqing as an example, as is shown in Figure 1, the treatment pool of it

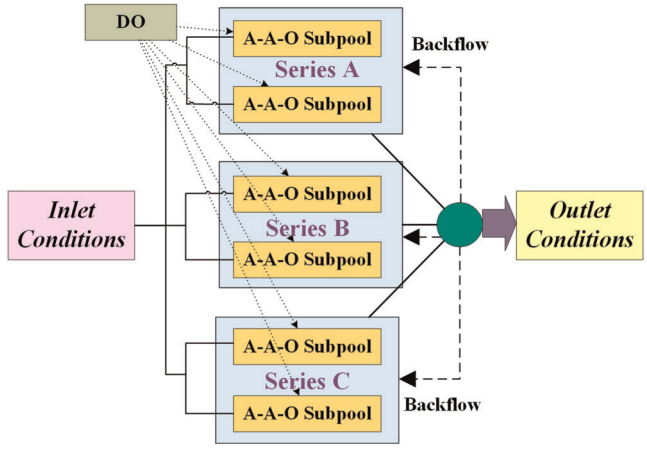


**FIGURE 1** Sketch map for infrastructures of a typical wastewater treatment plant. DO, dissolved oxygen

is divided into three series of pools: *A*, *B*, and *C*. And two specific treatment subpools can be further extracted from each series. Conventionally, treatment processes inside six subpools are viewed as those in an integrated treatment pool, neglecting internal correlations among processes of different subpools. Due to the constant backflow, treatment process of a subpool will certainly affect the following treatment effect of other subpools.

Hence, it is expected to tackle such a challenge by taking potential relevance among subpools into account. Fortunately, the newly emerged graph embedding (GE)[10,24] theory provides an appropriate solution for such demand. It fuses relations among entities into framework of high-order parallel computation, so that more resilient feature spaces can be obtained. The encoded feature can be further transferred into neural computing structure, yielding the graph neural network (GNN)[25] model. Therefore, this paper proposes graph embedding-based intelligent industrial decision for complex STP (GE-STP). First of all, GE is implemented to model latent linkages inside complex STP. On such basis, a GNN model is developed to realize the modeling of complex STP. Compared with existing technologies, the proposed GE-STP embedded idea of graph learning into the modeling of complex STP and is naturally a better solution for such purpose. To the best of our knowledge, this study creatively considers latent relations among subpools when employing intelligent computing to model complex STP. Main contributions of this paper are summarized as follows:

- It is recognized that potential relevance exists among subpools inside treatment plants, which is well worth investigating.
- The GE is introduced to capture latent relations inside complex STP, and a GNN model is developed as the intelligent computing method, forming the GE-STP.
- Simulative experiments are carried out on real-world scenes which derive from data of a treatment plant. It is proved that the performance of the GE-STP is about 6%–12% better than general neural network models.

The remainder of this paper is organized as follows. Section 2 introduces the problem scenarios and gives basic definitions. In Section 3, the detailed mathematical process of the

GE-STP is described in detail. Experimental settings, results, and analysis are displayed in Section 4. And we conclude this paper in Section 5.

## PROBLEM STATEMENT

As for the real-world sewage treatment plant that is investigated in this paper, the process structure is shown in Figure 1. It possesses totally six subpools whose index number is denoted as $i$. A complete set of IoT devices have been equipped with the plant, so that indicators of several major chemicals can be monitored in real time. The monitoring contents include three parts: inlet conditions, outlet results, and intermediate parameters. First of all, definitions of them are given as follows:

**Definition 1** (Inlet conditions). The initial pollutant indicators in the inlet point are defined as inlet conditions. Obviously, they need to be required during STP.

**Definition 2** (Outlet results). The pollutant indicators at the end of STP are defined as outlet results. Obviously, relative values between outlet results and inlet conditions reflect the treatment effect.

**Definition 3** (Intermediate parameters). The indicators of dissolved oxygen (DO) added into six subpools to reduce indicators of inlet pollutants are defined as intermediate parameters.

Given inlet conditions, investigation goal of this study is to predict outlet results in advance according to the amount of added intermediate parameters. Core pollutants inside inlet conditions and outlet conditions are ammonia nitrogen ($NH_3$-N) and chemical oxygen demand (COD). And materials involved in the intermediate parameters are DO. As for the proposed GE-STP, it manages to formulate a mapping from intermediate parameters to outlet results given inlet conditions. The main architecture of the GE-STP is shown in Figure 2. It contains three major procedures: GE, neural mapping, and training. Detailed process of the GE-STP is described as follows.

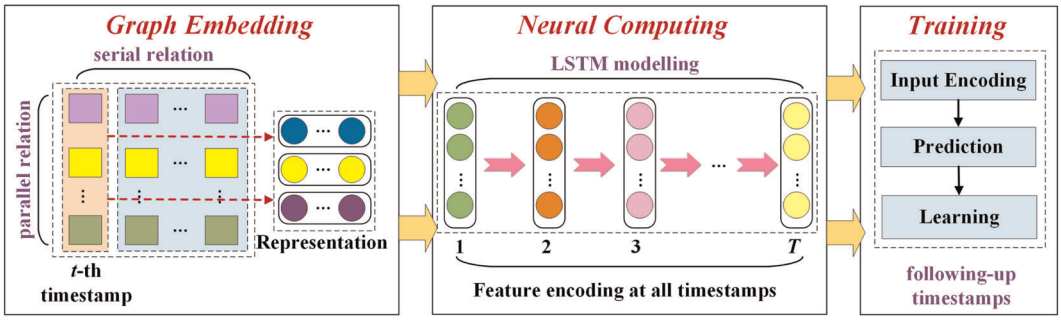


FIGURE 2 Main architecture of the proposed GE-STP. DO, dissolved oxygen; GE, graph embedding; LSTM, long short-term memory; STP, sewage treatment process

At the $t$th timestamp, inlet conditions with respect to six subpools are denoted as $x_i^{(t)}$, in which $i$ is the index number of subpools ranging from 1 to 6. For $x_i^{(t)}$, it has two types of relations: serial relations $R_i^{(t)}(\text{serial})$ and parallel relations $R_i^{(t)}(\text{parallel})$. The former exists between $x_i^{(t)}$ and other values inside $i$th subpool, and the latter exists between $x_i^{(t)}$ and other values of the $t$th timestamp. The two relation vectors can be concatenated into a vector
$R_i^{(t)(P)}$. In addition, six DO monitoring values at the $t$th timestamp constituting a vector $\Psi^{(t)}$. Then, $R_i^{(t)}(\text{serial})$ and $\Psi^{(t)}$ can be utilized together to generate a final representation for DO monitoring values: $\mathcal{X}^{(t)}$. Enumerating $t$ from 1 to $T$, all the DO monitoring values $x_i^{(t)}$ are viewed as a sequence with temporal characteristics. The long short-term memory (LSTM) is selected to model such a sequentially evolving process. Data of the first $T$ timestamps are used to train the prediction model. Having established the prediction model, outlet results at the following timestamps can be calculated according to inlet conditions and intermediate parameters. After that, an empirical error is introduced to construct an objective function, and Root Mean Square Prop (RMSProp) is selected as an optimization method to search optimal solutions.

## METHODOLOGY

This section gives the detailed mathematical descriptions of the algorithm workflow, containing three parts. Section 3.1 introduces the macroscopic process of the algorithm, Section 3.2 sets up a graph network to represent correlations inside multiple treatment subpools, and Section 3.3 designs a neural network architecture to finish the modeling of complex STP.

### Graph embedding

As shown in Figure 1, the backflow occurs at the end node near the outlet, deriving two aspects of occasional relations. First, intermediate processes of each timestamp are related to those of the following timestamps inside the same treatment subpool. Second, intermediate processes of each timestamp are related to those of the same timestamp across treatment subpools. As these links are latent and imperceptible, the links need to be estimated via sampling.

#### Serial relation

At the $t$th timestamp, let $x_i^{(t)}$ denote the DO value of the $i$th subpool. It is further assumed here that its influence is able to last during the following five timestamps. Accordingly, a sampling link can be generated inside such subpool from the $t$th timestamp to the $(t + 1)$th timestamp. Naturally, it is composed of totally six nodes. And the monitored DO values at these timestamps constitute a directed sampling link. All the monitoring values are nodes of the link and ranked in chronological order. Above all, the construction of each link undergoes five times of hops which refer to the transformations from one node into another one. Generalized to a directed link starting from the $t$th timestamp, it has five times of transformations from the node of the $t$th timestamp to the $(t + 1)$th, $(t + 2)$th, $(t + 3)$th, $(t + 4)$th, and $(t + 5)$th timestamps. As for each beginning node, it is required to implement multiple rounds of sampling to generate

multiple links. The index number of sampling rounds is denoted as $\tau$ which ranges from 1 to $\eta$. Taking the $\tau$th link as an example, each time of transformation is drawn from the following multinomial distribution:

$$Pr\left[N_{\tau,i}^{(t+1)}|N_{\tau,i}^{(t)}\right] = \begin{cases} |\phi_i^{\frac{1}{(t+1)}}|, & N_{\tau,i}^{(t+1)} \in |\phi_i^{(t+1)}|, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\phi_i^{(t+1)}$ denotes the set of possible nodes in the $t$th timestamp, and $|\cdot|$ counts the total number of it. After such sampling, the $\tau$th link which starts from $x_i^{(t)}$ is as the following format:

$$N_{\tau,i}^{(t)} \to N_{\tau,i}^{(t+1)} \to \quad \cdots \to N_{\tau,i}^{(t+5)}. \tag{2}$$

As all the nodes inside a link are sequential, their time-series relevance can be encoded. Let $n$ denote the index number of all the nodes inside sampling links, and $n$ ranges from 1 to 6. As for the $\tau$th link of $x_i^{(t)}$, hidden state of the transformation between two adjacent nodes is represented as

$$H_{\tau,i}^{(n\to n+1)} = \psi^{(n\to n+1)} \cdot N_{\tau,i}^{(n)} \cdot M^{(n\to n+1)}, \tag{3}$$

where $M^{(n\to n+1)}$ is the transition matrix between the $n$th node and the $(n+1)$th node, and $\psi^{(n\to n+1)}$ is the fading coefficient for the transformation between the $n$th node and the $(n+1)$th node. The $\psi^{(n\to n+1)}$ is computed as

$$\psi^{(n\to n+1)} = \exp\left(-\frac{7-n}{\xi}\right), \tag{4}$$

where $\xi$ is a parameter. It can be observed from the above formula that influence effect descends with the timestamps move forward. In other words, the nodes closer to the beginning node will receive greater influence. After six times of sampling operations, the relational representation for the $\tau$th link of $x_i^{(t)}$ is encoded as

$$R_{\tau,i}^{(t)}(\text{serial}) = \Delta_1\left\{W_{R1} \cdot \left[\sum_{n=1}^{5} \varepsilon_n \cdot H_{\tau,i}^{(n\to n+1)}\right] + b_{R1}\right\}, \tag{5}$$

where $\Delta_1(\cdot)$ is the rectified linear unit (ReLU) activation function, $W_{R1}$ is the weight parameter, and $b_{R1}$ is the bias parameter. The $n$ ranges from 1 to 5, which corresponds to the five times of transformation inside each sampling link. After all the $\mu$ rounds of sampling operations, a total representative vector for serial relation of $x_i^{(t)}$ can be deduced as

$$R_i^{(t)}(\text{serial}) = \sum_{\tau=1}^{\mu} \gamma_\tau \cdot R_{\tau,i}^{(t)}(\text{serial}), \tag{6}$$

where $\gamma_\tau$ is the weight for $R_{\tau,i}^{(t)}$ and satisfies the following condition:

$$\sum_{\tau=1}^{\mu} \gamma_\tau = 1. \tag{7}$$

### Parallel relation

The latent relations not only contain sequential types, but also have parallel ones. At the $t$th timestamp, DO monitoring values of each in each subpool have correlations with those inside other subpools. Let $j$ denote the index number of all the subpools except $i$. Enumerating $i$ and $j$ from 1 to 6, $x_i^{(t)}$ and $x_j^{(t)}$ denote all pairs of DO values which belong to two different subpools. As no sequential relations exist between a pair of $x_i^{(t)}$ and $x_j^{(t)}$, the relation degree between them is likely to possess remarkable randomness. As for $x_i^{(t)}$, it is likely to have correlations with five other monitoring values of such timestamp which belongs to other five subpools. Above all, it is expected to model such random relations between all pairs of $x_i^{(t)}$ and $x_j^{(t)}$.

Gaussian distribution is selected to describe such randomness, and two different distributions are set up in accordance with two different situations. As shown in Figure 1, all the six subpools are divided into three different series: $A$, $B$, and $C$. Naturally, each series has two subpools. Let $\delta^{(t)}(i,j)$ denote random relation degree between $x_i^{(t)}$ and $x_j^{(t)}$. As for a pair of $x_i^{(t)}$ and $x_j^{(t)}$ which belong to the same series, the relations between them are drawn from the Gaussian distribution with mean value $\mu_1$ and variance $\sigma_1^2$, which can be represented as $\delta^{(t)}(i,j) \sim \mathcal{N}_1(\mu_1, \sigma_1^2)$. As for a pair of $x_i^{(t)}$ and $x_j^{(t)}$ which belong to two different series, the relations between them are drawn from the Gaussian distribution with mean value $\mu_2$ and variance $\sigma_2^2$, which can be represented as $\delta^{(t)}(i,j) \sim \mathcal{N}_2(\mu_2, \sigma_2^2)$. Considering the relations inside series and across series, the $\mu_1$ ranges from 0.5 to 1, and the $\mu_2$ ranges from 0 to 0.5.

As for $x_i^{(t)}$, a set of sampling operations are utilized to measure relation degree between it and all the other $x_j^{(t)}$. Let $c$ denote the index number of all the $C$ sampling operation rounds. During the $c$th round of sampling operation, representative vector for relation degree between $x_i^{(t)}$ and $x_j^{(t)}$ is measured as

$$R_{c,i}^{(t)}(\text{parallel}) = \left[\delta^{(t)}(i,j)\right]_{j=1,2,\ldots,6; j\neq i}. \tag{8}$$

Making the $i$ constant, the aforementioned $R_{c,i}^{(t)}(\text{parallel})$ has five elements corresponding to relation degree values between the $x_i^{(t)}$ and other five values of $x_j^{(t)}$. After all the $C$ rounds of sampling operations, the total representative vector for parallel relations of $x_i^{(t)}$ is represented as

$$R_i^{(t)}(\text{parallel}) = \sum_{c=1}^{C} \gamma_c \cdot R_{c,i}^{(t)}(\text{parallel}), \tag{9}$$

where $\gamma_\tau$ is the weight for $R_{\tau,i}^{(t)}$ and satisfies the following condition:

$$\sum_{c=1}^{C} \gamma_c = 1. \tag{10}$$

## Neural computing

The complex STP can be regarded as a set of graph networks, in which nodes refer to monitored DO values, and edges refer to serial linkages and parallel linkages among monitoring values. Representation for nodes derives from their initial monitoring values, and representation for edges derives from latent linkages inside complex STP. This subsection first integrates basic

graph-level feature representations, and then put them into a neural network structure for computation.

## Feature integration

At the $t$th timestamp, the proposed GE-STP is to formulate a mapping from monitoring values to the outlet results given inlet conditions. The total representative matrix for input values at such timestamp is denoted as $\mathcal{X}^{(t)}$. The $\mathcal{X}^{(t)}$ takes the following form:

$$\mathcal{X}^{(t)} = \left[\mathcal{X}_{x,i}^{(t)}, \mathcal{X}_{R,i}^{(t)}\right]_{i=1,2,\ldots,6}, \tag{11}$$

where $i$ is the index number of subpools and ranges from 1 to 6. It is a type of matrix with six lines which correspond to monitoring values of six subpools. The $\mathcal{X}^{(t)}$ contains two main parts: $\mathcal{X}_{x,i}^{(t)}$ and $\mathcal{X}_{R,i}^{(t)}$. The former part refers to encoded factor for monitoring values at such timestamp, and the latter part refers to encoded factor for their relation features. It is noted that $\mathcal{X}_{R,i}^{(t)}$ is derived from serial relations and parallel relations of $x_i^{(t)}$.

The mentioned two relations $R_i^{(t)}$(serial) and $R_i^{(t)}$(parallel) can be first integrated into a temporal representative vector $R_i^{(t)}$(temp) as

$$R_i^{(t)}(\text{temp}) = R_i^{(t)}(\text{parallel}) \cdot w_1 + R_i^{(t)}(\text{serial}) \cdot w_2, \tag{12}$$

where $w_1$ and $w_2$ are the two sliding matrices that match different dimensions of two vectors. To obtain a comprehensive representation for relations of $x_i^{(t)}$, an iterative propagation process is required. It is a multiround iterative process and represented as

$$R_i^{(t)(p+1)} = \Delta_1\left[W_{R2} \cdot R_i^{(t)(p)} + b_{R2}\right], \tag{13}$$

where $W_{R2}$ is the weight parameter, $b_{R2}$ is the bias parameter, and $p$ is the index number of iterative rounds that range from 1 to $P$. At the initial status where $p = 0$, $R_i^{(t)(p)}$ is equal to $R_i^{(t)}$(temp). And after all the $P$ rounds of iterations, the obtained $R_i^{(t)(p)}$ is denoted as $R_i^{(t)(P)}$.

As for all the six monitoring values at the $t$th timestamp, they can be concatenated into a six-dimensional vector $\Psi^{(t)}$. So far, initial representation for nodes and edges in graph network at the $t$th timestamp has been obtained. Among, representation for nodes is the vector of six monitoring values $\Psi^{(t)}$, and representation for edges is the relation matrix $R_i^{(t)(P)}$. It is pointed that both nodes and edges never exist independently, and that they are highly correlated with other. Thus, a cross-iteration operation is added to the feature integration process. Specifically, the mentioned $\Psi^{(t)}$ and $R_i^{(t)(P)}$ are, respectively, updated once to finally generate two parts of $\mathcal{X}^{(t)}$: $\mathcal{X}_{x,i}^{(t)}$ and $\mathcal{X}_{R,i}^{(t)}$. The obtainment of $\mathcal{X}_{x,i}^{(t)}$ is determined by both itself and $\mathcal{X}_{R,i}^{(t)}$. And similar to $\mathcal{X}_{R,i}^{(t)}$, its obtainment is determined by both itself and $\mathcal{X}_{x,i}^{(t)}$. Such two cross-iteration processes can be represented as

$$\mathcal{X}_{x,i}^{(t)} = w_3 \cdot \Psi^{(t)} \cdot w_4 + \lambda_1 \cdot R_i^{(t)(P)}, \tag{14}$$

$$\mathcal{X}_{R,i}^{(t)} = \lambda_2 \cdot R_i^{(t)(P)} + w_5 \cdot \Psi^{(t)} \cdot w_6, \tag{15}$$

where $w_3$, $w_4$, $w_5$, and $w_6$ are the four vectors or matrices that are used to match dimensions between $\mathcal{X}_{x,i}^{(t)}$ and $\mathcal{X}_{R,i}^{(t)}$, and $\lambda_1$ and $\lambda_2$ are two trade-off parameters to set weight for $R_i^{(t)(P)}$. The $R_i^{(t)(P)}$ is with the form of matrix, while the $\Psi^{(t)}$ is with the form of vector. Thus, $w_3$, $w_4$, $w_5$, and

$w_6$ are responsible for transforming low-dimensional $\Psi^{(t)}$ into high-dimensional latent matrices, so that dimension of $\Psi^{(t)}$ matches that of $R_i^{(t)(P)}$.

## Dependency modeling

Having deduced $X_{x,i}^{(t)}$ and $X_{R,i}^{(t)}$ via the above procedures, the total representative matrix for input values at the $t$th timestamp, $\mathcal{X}^{(t)}$, is obtained. At the same time, STP is a certainly complex, as well as a sequentially evolving process. Enumerating $t$ from 1 to $T$, temporal dependency exists among all of the values of $\mathcal{X}^{(t)}$. To model such dependency, recurrent neural network (RNN) models can be utilized for this purpose. The LSTM model is a classical variant of RNN, and was specially developed for modeling long-term dependency characteristics inside complex industrial systems. Due to the excellent ability to model long-term complex processes, the LSTM has been put into realistic practice in many industrial or commercial applications. Structure of the LSTM model is composed of three gates: input gate (InG), forgetting gate (FoG), and output gate (OuG). At the $t$th timestamp, the InG controls the degree that the input variable $I(t)$ is saved into major cell state $S(t)$. The FoG controls the degree that the major cell state at the $(t-1)$th timestamp is retained to the major cell state at the current timestamp. The OuG controls the degree that major cell state $S(t)$ is saved into final output $O(t)$.

As for the FoG, control factor of it at the $t$th timestamp can be expressed as the following formula:

$$F(t) = \Delta_2\{W_F \cdot [O(t-1) \oplus I(t)] + b_F\}, \tag{16}$$

where $W_F$ and $b_F$ are the weight parameter and bias parameter for the connection between InG and FoG, $\Delta_2(\cdot)$ is the sigmoid activation function, and the input variable at the $t$th timestamp is as

$$I(t) = \mathcal{X}^{(t)}. \tag{17}$$

Due to the fact that range for output of the sigmoid function is $(0, 1)$, the $F(t)$ is a real number that ranges from 0 to 1. The $F(t)$ is equal to 0 when historical information is completely forgotten, and is equal to 1 when none of the historical information is forgotten.

As for the InG, control vector of it at the $t$th timestamp is deduced as

$$E(t) = \Delta_2\{W_{E1} \cdot [O(t-1) \oplus I(t)] + b_{E1}\}, \tag{18}$$

where $W_{E1}$ and $b_{E1}$ are the weight parameter and bias parameter of the connection between InG and OuG. Representation for cell state at the $t$th timestamp can be deduced as the following formula:

$$S(t) = F(t) \cdot S(t-1) + E(t) \cdot \widetilde{S}(t), \tag{19}$$

where $\tilde{S}(t)$ is a temporal vector defined as follows:

$$\widetilde{S}(t) = \Delta_3\{W_{E2} \cdot [O(t-1) \oplus I(t)] + b_{E2}\}, \tag{20}$$

where $W_{E2}$ and $b_{E2}$ are the bias parameter and bias parameter of the connection between cell state and OuG, and $\Delta_3(\cdot)$ is the tanh activation function.

As for the OuG, control vector of it at the $t$th timestamp is deduced as

$$D(t) = \Delta_2\{W_D \cdot [O(t-1) \oplus I(t)] + b_D\}, \tag{21}$$

where $W_D$ and $b_D$ are the bias parameter and bias parameter for output vector of OuG. Output of the OuG is the output of the LSTM model, and can be calculated as

$$O(t) = D(t) \cdot \Delta_3[S(t)]. \tag{22}$$

## Training

Having undergone all the $T$ rounds of evolvement processes, a model can be trained for predicting treatment results of the following timestamps. Monitoring data at the first $T$ timestamps are utilized to train prediction models for treatment processes following timestamps. Input of the prediction model is monitored DO values in six subpools, and output of the prediction model is the treatment effect. It is noted that the treatment effect is not absolute outlet results. Instead, it is supposed to be quantification for relative reduction compared with inlet conditions. At each timestamp, the output of the prediction model is actually the quotient of outlet results divided by inlet conditions. Taking the $(t+1)$ timestamp as an example, treatment effect at such timestamp can be represented as $O(t)$. To map the $O(t)$ into higher-level forms, two multilayer perception (MLP) networks are introduced for this purpose, leading to the following two formulas:

$$Y_1(T+1) = \Delta_1\{MLP_1[O(T+1)]\}, \tag{23}$$

$$Y_2(T+1) = \Delta_4\{MLP_2[O(T+1)]\}, \tag{24}$$

where $MLP_1(\cdot)$ and $MLP_2(\cdot)$ are the two MLP operators for feature transformation, and $\Delta_4(\cdot)$ is the leaky ReLU activation function. The two obtained $Y_1(T+1)$ and $Y_2(T+1)$ are two matrices, representing two variants of $O(T+1)$ in terms of two perspectives.

After that, the $Y_1(T+1)$ and $Y_2(T+1)$ can be concatenated into a total matrix by combining the product of them two, which can be expressed as

$$A(T+1) = Y_1(T+1) \cdot [Y_2(T+1)]^T. \tag{25}$$

And the $Z(T+1)$ can be further mapped into predicted result through a linear neural mapping procedure, which can be expressed as

$$\hat{Z}(T+1) = \Delta_1[W_Z \cdot A(T+1) + b_Z], \tag{26}$$

where $W_Z$ is the weight parameter, $b_Z$ is the bias parameter, and $\hat{Z}(T+1)$ is the estimated result at the $(T+1)$th timestamp. Empirical error-based training objective function can be formulated as the following $L_2$-norm formula:

$$\min\left\{\sum_{i=1}^{6}\sum_{t=1}^{T}\left[\lambda_3 \cdot \left\|Z(\hat{T}+1) - Z(T+1)\right\|_F^2 + \lambda_4 \cdot \|\Theta\|_F^2\right]\right\}, \tag{27}$$

where $\lambda_3$ and $\lambda_4$ are the two trade-off parameters. And the RMSProp approach can be selected as the optimizer to find the optimal solution for the above formula. After learning, the complete

prediction model is successfully formulated. Given inlet conditions at any following-up timestamps, outlet results can be calculated according to the input of DO values inside six subpools.

## EXPERIMENTS AND ANALYSIS

### Data preparation

Having formulated technical methods named GE-STP in the previous parts, experiments on a real-world data set are required to evaluate the proposed GE-STP. The real-world data set was collected from a realistic sewage treatment plant that is located in Nan'an District, Chongqing, China.[11] This treatment plant has been equipped with IoT devices to monitor business values during industrial processes, so that digital management mode can be realized. The devices have been running steadily since 2018, and monitoring data were collected from July 1, 2018 to June 30, 2019. Inside each natural day, the IoT devices automatically implemented about 200–300 times of data monitoring. Each time of monitoring produces one piece of monitoring data, and monitoring frequency varies with different days. To deal with such frequency inconsistence, the first 200 pieces of data inside each day are uniformly selected as the monitoring data of each day. Generalized to the period of 1 year, more than seventy thousand pieces of data can be used to construct the experimental data set. It is further assumed that sequential characteristics exist among all of the data, so that they can be modeled as a sequentially evolving process.

As the data set came from a sewage treatment plant in Chongqing, data structure completely corresponds to process structure of the plant. For each piece of data, it contains 10 monitoring values. The first two values are inlet conditions with respect to two pollutants, and the last two values are outlet results with respect to two pollutants. Obviously, the middle six values are monitored intermediate parameters with respect to six subpools. In addition, it is expected to visualize the data distribution of all of the 10 volumes of data. Figure 3

| *No.* | *Symbol* | *Definition* | *Min* | *Max* | *Mean* | *S.D.* |
|---|---|---|---|---|---|---|
| 1 | $\alpha_1^{(t)}$ | Inlet NH3-N value at the $t$-th timestamp | 9.339 | 1061.544 | 441.008 | 195.165 |
| 2 | $\alpha_2^{(t)}$ | Inlet COD value at the $t$-th timestamp | 0.156 | 110.467 | 27.283 | 9.889 |
| 3 | $x_1^{(t)}$ | Intermediate DO value inside 1st subpool | 1.002 | 9.562 | 2.810 | 1.519 |
| 4 | $x_2^{(t)}$ | Intermediate DO value inside 2nd subpool | 1.000 | 9.287 | 3.293 | 1.861 |
| 5 | $x_3^{(t)}$ | Intermediate DO value inside 3rd subpool | 1.001 | 9.413 | 2.607 | 1.117 |
| 6 | $x_4^{(t)}$ | Intermediate DO value inside 4th subpool | 1.000 | 9.088 | 2.691 | 1.109 |
| 7 | $x_5^{(t)}$ | Intermediate DO value inside 5th subpool | 1.003 | 9.956 | 5.571 | 2.604 |
| 8 | $x_6^{(t)}$ | Intermediate DO value inside 6th subpool | 1.241 | 9.973 | 6.152 | 3.118 |
| 9 | $\beta_1^{(t)}$ | Outlet NH3-N value at the $t$-th timestamp | 3.016 | 49.475 | 25.606 | 0.317 |
| 10 | $\beta_2^{(t)}$ | Outlet COD value at the $t$-th timestamp | 0.156 | 28.607 | 2.215 | 0.703 |

**FIGURE 3** Definition and statistical characteristics of the experimental data set. COD, chemical oxygen demand; DO, dissolved oxygen; SD, standard deviation

demonstrates both definitions and statistical characteristics for all the symbols, in which three types of symbols are included. First, $\alpha_1^{(t)}$ and $\alpha_2^{(t)}$, respectively, denote inlet conditions in terms of two major pollutants at the $t$th timestamp. Second, $\beta_1^{(t)}$ and $\beta_2^{(t)}$, respectively, denote outlet results in terms of two major pollutants at the $t$th timestamp. Third, $x_i^{(t)}(1,\ 2, ..., 6)$ denotes six monitored DO values inside six subpools. There are also some abbreviated terms involved in Figure 3. Enumerating $t$ from 1 to $T$, "Min" denotes the minimum value, "Max" denotes the maximum value, "Mean" denotes the mean value, and "*SD*" denotes the standard deviation value. It can be seen from Figure 3 that minimum values, maximum values, and mean values of each type of variables remain relatively steady. Taking $x_i^{(t)}(1,\ 2, ..., 6)$ as an example, their minimum values range from 1 to 1.2, their mean values range from 2 to 6, and their maximum values range from 9 to 10. And for all of the variables, their standard deviation values remain relatively low compared with mean values. In all, the experimental data set is evenly distributed, so that data analysis algorithms are able to be implemented reasonably.

## Experimental settings

To assess the main performance of the proposed GE-STP, some basic metrics are required for this purpose. As the research object in this study is actually a regression problem, two relevant metrics are selected for this purpose: mean average error (MAE) and root mean square error (RMSE). The two metrics are defined as the following two formulas:

$$MAE = \frac{1}{U}\sum_{u=1}^{U}|y_u - \hat{y}_u|, \tag{28}$$

$$RMSE = \sqrt{\frac{1}{U}\sum_{u=1}^{U}(y_u - \hat{y}_u)^2}, \tag{29}$$

where $\hat{y}_u$ is the predicted value, $y_u$ is the real value, and $u$ is the index number of samples that range from 1 to $U$. According to definitions, MAE and RMSE measure the difference between predicted values and real values.

To achieve the comparison effect, some typical methods that had been utilized to model STP are selected as baselines, which are briefly described as follows:

*CNN*—It refers to the convolutional neural network model which is a kind of feed-forward neural network with deep structure and convolution computation.[11]

*LSTM*—It refers to the LSTM model which is a kind of time-aware neural network to deal with long-term dependency.[25]

*GRU*—It refers to the gated recurrent unit model which modifies the gated structure of LSTM and is a variant of the LSTM model.[12]

*FNN*—It refers to the fuzzy neural network model which integrates fuzzy logic into neural networks to endow it with the ability of inference.[13]

More detailed descriptions of these four comparison methods can be found in corresponding literatures. The proposed GE-STP and other baselines were implemented via the programming language Python and with the aid of a famous deep learning tool TensorFlow.* The hardware environment lies in a deep learning workstation with 28-core CPU and a GPU (RTX-2080Ti). The number of sampling rounds $\eta$ is set to 10, the fading parameter $\xi$ in Equation (4) is set to 10, $\mu_1$ and $\sigma_1^2$ are set to 0.3 and 0.04, $\mu_2$ and $\sigma_2^2$ are set to 0.8 and 0.03, the

number of sampling rounds $C$ is set to 10. $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$ are all set to 0.5. Proportion of training data is set to 60% initially, and will change multiple times during experiments. Learning rate of the GE-STP is set to 0.001 initially, and will change multiple times during experiments.

## Results and analysis

Retaining proportion of training data as the level of 60%, performance of GE-STP and baseline methods is evaluated in terms of COD and $NH_3$-N with the use of two metrics: MAE and RMSE. Collaborative results of MAE and RMSE for outlet COD with different proportions of learning rate are illustrated in Figure 4. It is composed of four subfigures corresponding to MAE and RMSE results under four learning rate values: 0.001, 0.003, 0.005, and 0.01. Inside each subfigure, the $X$-axis refers to a range of MAE values and the $Y$-axis refers to a range of RMSE values. Each scatter inside a subfigure refers to a pair of "MAE-RMSE" values acquired by one method. As MAE and RMSE measure the distance between real values and predicted values, the scatters closer to the origin reflects a better prediction effect. Figure 5 demonstrates MAE and RMSE results for outlet $NH_3$-N under

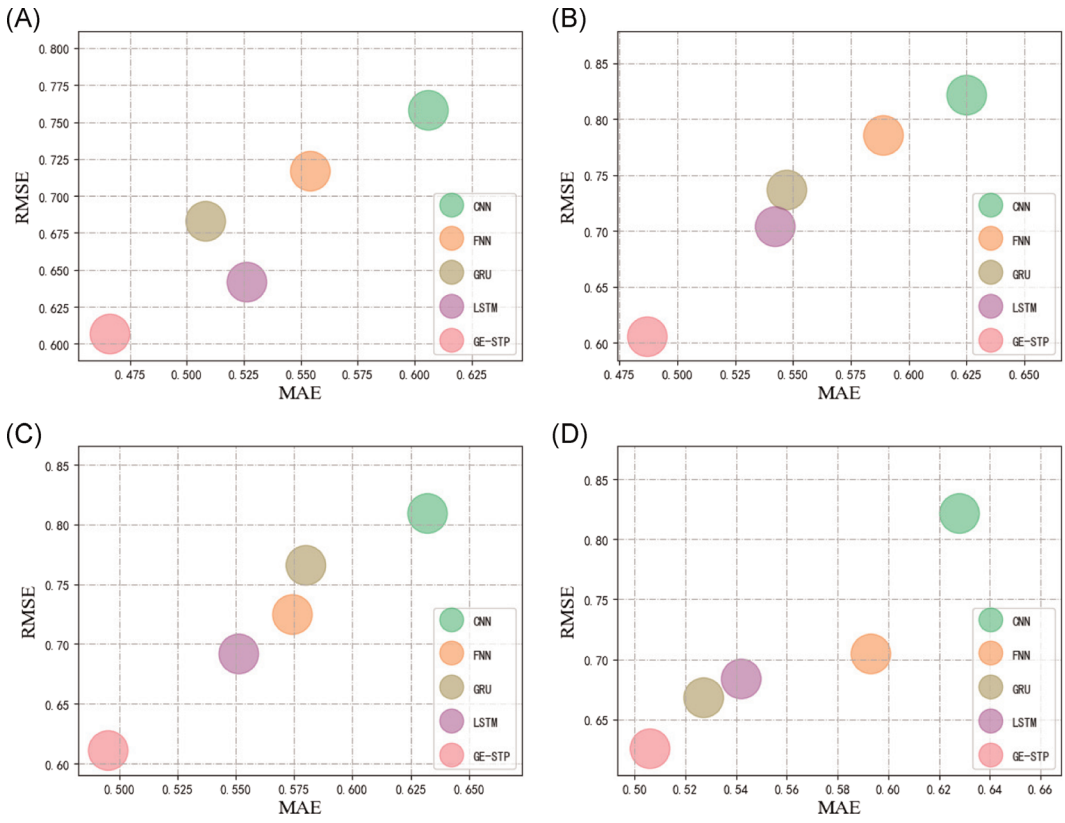


**FIGURE 4** MAE and RMSE results concerning outlet COD with respect to different learning rates: (A) 0.001, (B) 0.003, (C) 0.005, and (D) 0.01. COD, chemical oxygen demand; MAE, mean average error; RMSE, root mean square error

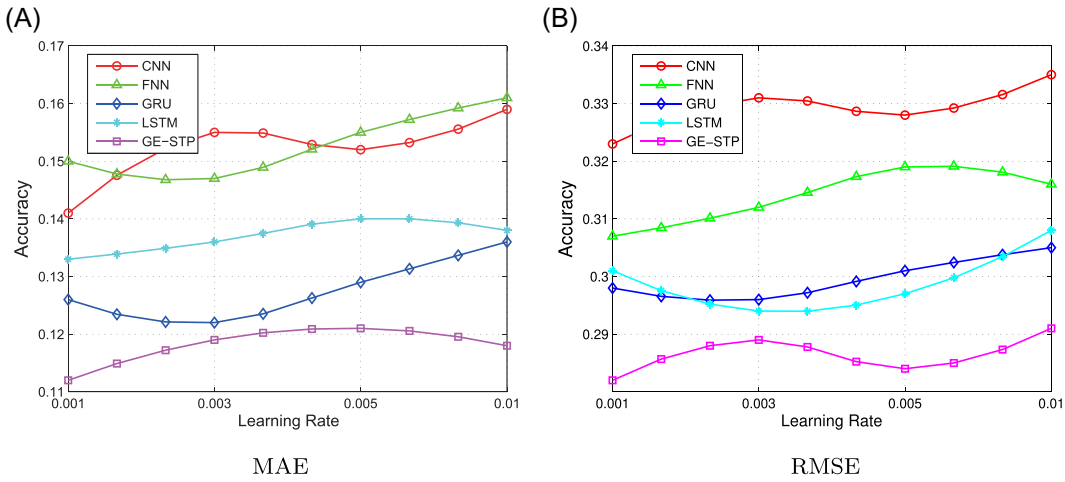


**FIGURE 5** (A) MAE and (B) RMSE results concerning outlet $NH_3$-N with respect to different learning rates. CNN, convolutional neural network; FNN, fuzzy neural network; GE, graph embedding; GRU, gated recurrent unit; LSTM, long short-term memory; MAE, mean average error; RMSE, root mean square error; STP, sewage treatment process

four learning rate values: 0.001, 0.003, 0.005, and 0.01. It is composed of two subfigures corresponding to MAE results and RMSE results, respectively. Inside each subfigure, the *X*-axis refers to learning rate values that range from 0.001 to 0.01, the *Y*-axis refers to values of metrics corresponding to experimental methods. As MAE and RMSE measure the distance between real values and predicted values, lower curves reflect the better performance of the corresponding methods compared with others. It can be observed from these figures that the proposed GE-STP is able to obtain the best prediction performance under all of the situations. Taking MAE results of outlet COD as an example, the GE-STP is about 5% better than LSTM, 6% better than GRU, 9% better than FNN, and 11% better than CNN. In subfigures of Figure 4, the scatters corresponding to GE-STP are always closer to the origin than others. In subfigures of Figure 5, the curves corresponding to GE-STP are always located below other curves.

Retaining learning rate value as the level of 0.001, performance of GE-STP and baseline methods is evaluated under two proportions of training data: 60% and 70%. In this group of experiments, MAE and RMSE results with respect to outlet COD and outlet $NH_3$-N are illustrated in Figures 6 and 7. Both of them have two subfigures corresponding to results under two different training sizes: 60% and 70%. Each subfigure has two clusters of bars which correspond to MAE results and RMSE results, respectively. Inside each subfigure, the *X*-axis lists name of two clusters, and the *Y*-axis denotes metric values. It can be seen from these figures that the proposed GE-STP always performs better than others under different scenario settings. During the above two groups of experiments, excellent performance of GE-STP is well acknowledged through three types of visualized figures: scatter diagram, curve diagram, and bar diagram. The obtainment of the above results can be attributed as two aspects of reasons. First, the GE scheme is introduced when encoding initial features into higher-level forms, which are distinguished from others. The introduction of GE is able to help the GE-STP extract more fine-grained features, contributing a lot to performance promotion. Second, the fusion of GE and neural networks is able to further improve the ability of prediction and calculation. The above

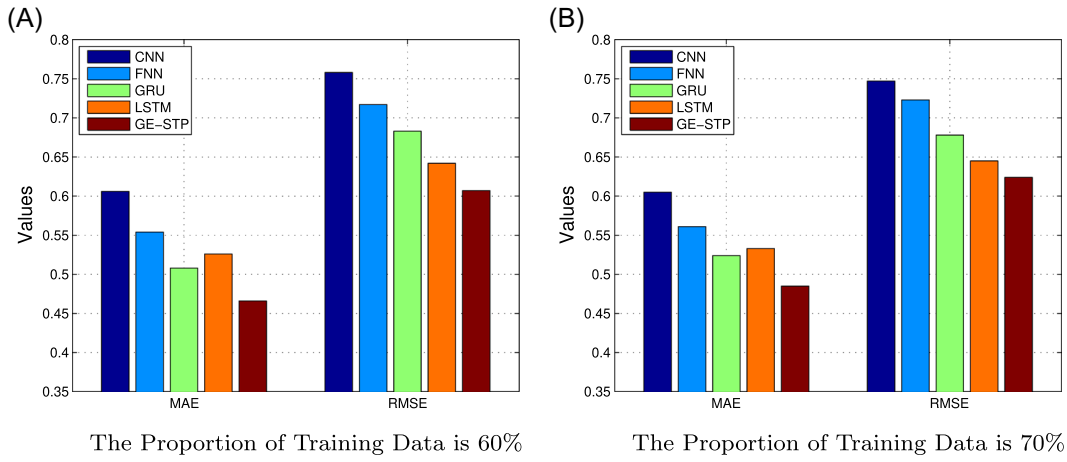


**FIGURE 6** MAE and RMSE results concerning outlet COD with respect to two training sizes: The proportion of training data is (A) 60% and (B) 70%. COD, chemical oxygen demand; CNN, convolutional neural network; FNN, fuzzy neural network; GE, graph embedding; GRU, gated recurrent unit; LSTM, long short-term memory; MAE, mean average error; RMSE, root mean square error; STP, sewage treatment process

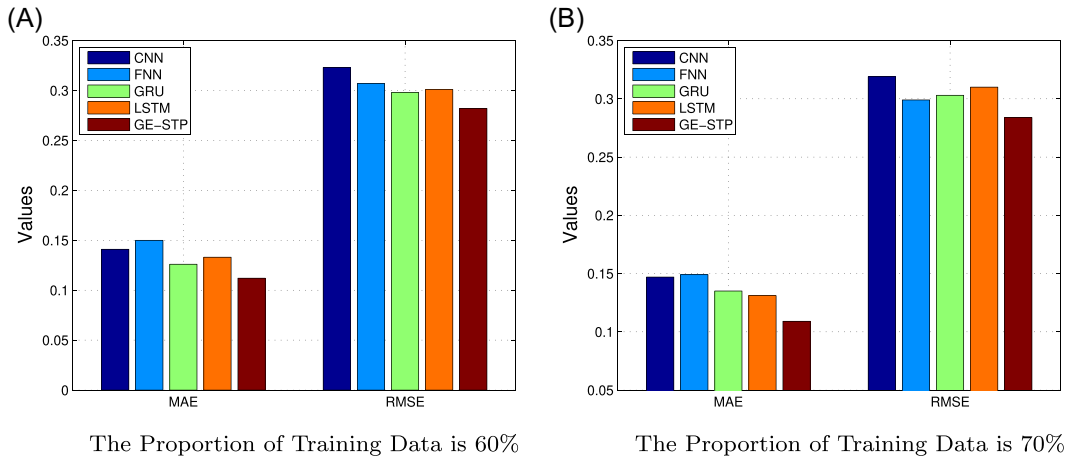


**FIGURE 7** MAE and RMSE results concerning outlet NH3-N with respect to two training sizes: The proportion of training data is (A) 60% and (B) 70%. CNN, convolutional neural network; FNN, fuzzy neural network; GE, graph embedding; GRU, gated recurrent unit; LSTM, long short-term memory; MAE, mean average error; RMSE, root mean square error; STP, sewage treatment process

two aspects of reasons are likely to make the GE-STP more effective than general neural network models.

Having assessed the efficiency of the proposed GE-STP, it is also expected to conduct another group of experiments to evaluate robustness of it. This group of experiments contains multiple times of small experiments. Inside each small experiment, a pair of parameters will

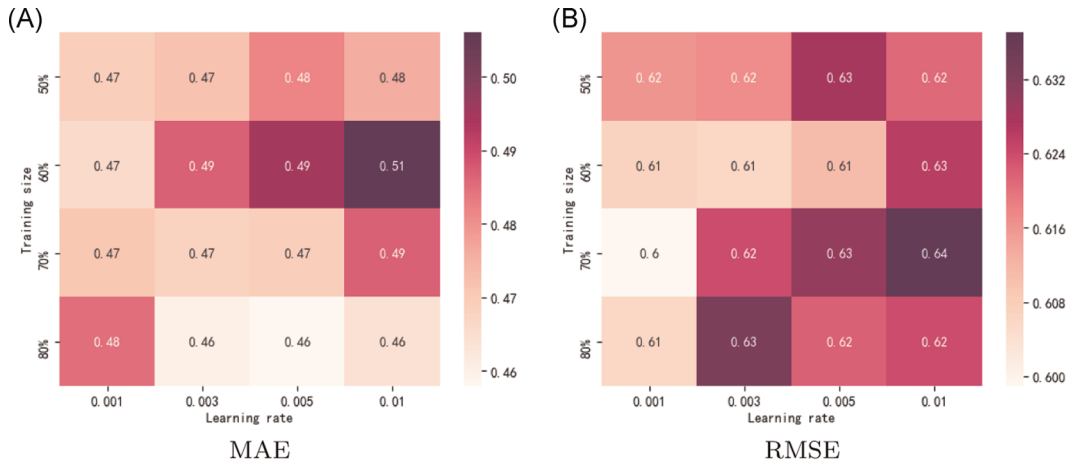


**FIGURE 8** Parameter sensitivity results concerning outlet COD with respect to two metrics: (A) MAE and (B) RMSE. COD, chemical oxygen demand; MAE, mean average error; RMSE, root mean square error

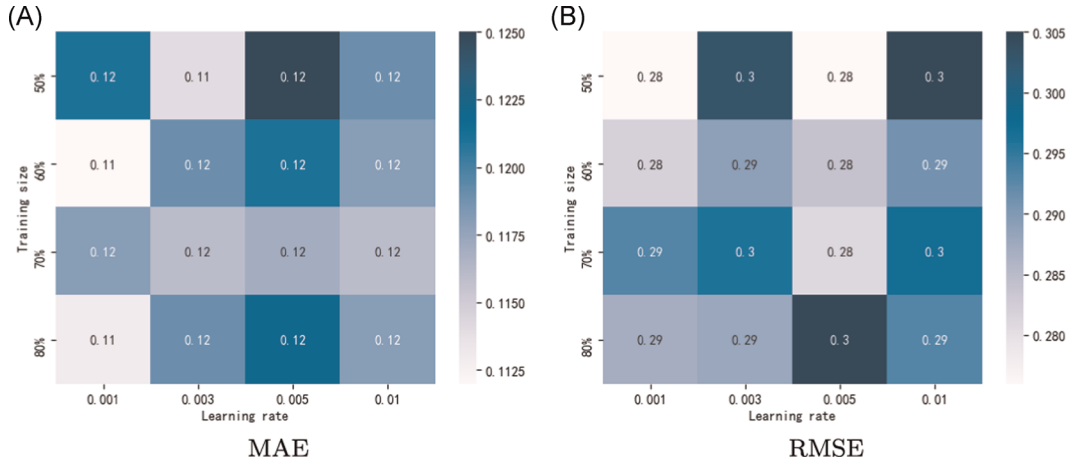


**FIGURE 9** Parameter sensitivity results concerning outlet $NH_3$-N with respect to two metrics: (A) MAE and (B) RMSE. MAE, mean average error; RMSE, root mean square error

change multiple times. Under such setting, performance fluctuation of GE-STP is investigated. If performance of it never fluctuates with the change of parameters, the GE-STP possesses good stability. The two parameters selected in this group of experiments are learning rate and proportion of training data. Note that the performance of the GE-STP is investigated singly, without comparing with any baselines. And evaluation metrics in this group of experiments are also MAE and RMSE that have been utilized before. Parameter sensitivity results of GE-STP with respect to outlet COD and outlet $NH_3$-N are illustrated in Figures 8 and 9. Both of them have two subfigures corresponding to two metrics: MAE and RMSE. Inside each subfigure, the *X*-axis refers to learning rate values ranging from 0.001 to 0.01, and the *Y*-axis refers to training size values ranging from 50% to 80%. The blocks inside each subfigure refer to metric values

under a pair of parameter settings. It can be observed from these figures that the performance of the GE-STP never strongly fluctuates with the change of parameter settings. This phenomenon indicates that the proposed GE-STP possesses proper robustness, because it is not susceptible to parameter changes. A possible explanation for this lies in the fact that the GE-STP manages to improve the depth of feature spaces by introducing the GE scheme, which improves the stability of GE-STP to a large extent.

To sum up, the above three groups of experiments are able to evaluate the excellent performance of the GE-STP. It not only possesses good efficiency, but also possesses proper robustness.

## CONCLUSION

Due to the significant role to sustainable development, the investigation towards STP has gained worldwide attention. And in the era of industry 4.0, newly emerged artificial intelligence technology has provided more innovative insights into this domain. In this context, intelligent algorithms-driven industrial decision systems have been a general demand for modeling complex STP. Existing researches modeled complex STP with the use of various neural network models, yet neglecting the fact that latent and occasional relations exist inside complex STP. To bridge such gap, this paper proposes a novel approach named GE-STP. The GE scheme is employed to enhance feature extraction and neural computing structure is utilized to simulate uncertain biochemical transformation inside STP. The introduction of GE can not only improves the fineness of feature spaces, but also improves the representative ability of models towards complex industrial processes. At last, a set of experiments are conducted on a real-world data set to evaluate the performance of the GE-STP. Four typical neural network models are selected as baseline methods. Three groups of experiments prove that the GE-STP properly possesses both efficiency and robustness.[14–21]

### ACKNOWLEDGMENTS

This work was supported in part by Chongqing Natural Science Foundation of China under Grant cstc2019jcyj-msxmX0747, in part by National Language Commission Research Program of China under Grant YB135-121, in part by the Science and Technology Research Project of Chongqing Municipal Education Commission under Grant KJZD-M202000801, in part by Innovation Group of New Technologies for Industrial Pollution Control of Chongqing Education Commission under Grant CXQT19023, in part by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) (JP18K18044, JP21K17736), and in part by the Key Research Project of Chongqing Technology and Business University under Grant ZDPTTD201917 and ctbuyqzx08.

### CONFLICT OF INTERESTS

The author declares that there is no conflict of interests.

## REFERENCES


1. Liu S, Pan Z, Fu W, Cheng X. Fractal generation method based on asymptote family of generalized Mandelbrot set and its application. *J Nonlinear Sci Appl*. 2017;10(3):1148-1161.
2. Su J, Yang Y, Yang T. Measuring knowledge diffusion efficiency in R&D network. *Knowl Manage Res Pract*. 2018;16(2):208-219.
3. Li H, Chen X, Guo Z, Xu J, Shen Y, Gao X. Data-driven peer-to-peer blockchain framework for water consumption management [published online ahead of print April 2, 2021]. *Peer Peer Netw Appl*. https://doi.org/10.1007/s12083-021-01121-6
4. Guo Z, Tang L, Guo T, Yu K, Alazab M, Shalaginov A. Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace. *Future Gener Comput Syst.* 2021;117:205-218.
5. Liu Z, Huang Y, Li J, Cheng X, Shen C. DivORAM: towards a practical oblivious RAM with variable block size. *Inf Sci*. 2018;447:1-11.
6. Guo Z, Shen Y, Aloqaily M, Jararweh Y, Yu K. Probabilistic inference-based modeling for sustainable environmental systems under hybrid cloud infrastructure. *Simul Model Pract Theory*. 2021;107:102215.
7. Krueger M, Luo H, Ding SX, Dominic S, Yin S. Data-driven approach of KPI monitoring and prediction with application to wastewater treatment process. *IFAC-PapersOnLine*. 2015;48(21):627-632.
8. Qiao J, Li F, Han H, Li W. Constructive algorithm for fully connected cascade feedforward neural networks. *Neurocomputing*. 2016;182(1):154-164.
9. Huang M, Tian D, Liu H, et al. A hybrid fuzzy wavelet neural network model with self-adapted fuzzy-means clustering and genetic algorithm for water quality prediction in rivers. *Complexity*. 2018;2018:1-11.
10. Zhang J, Yu K, Wen Z, Qi X, Paul AK. 3D reconstruction for motion blurred images using deep learning-based intelligent systems. *Comput Mater Contin*. 2021;66(2):2087-2104. https://doi.org/10.32604/cmc.2020.014220
11. Guo Z, Du B, Wang J, et al. Data-driven prediction and control of wastewater treatment process through the combination of convolutional neural network and recurrent neural network. *RSC Adv*. 2020;10:13410-13419.
12. Zeng W, Guo Z, Shen Y, et al. Data-driven management for fuzzy sewage treatment processes using hybrid neural computing [published online ahead of print January, 2021]. *Neural Comput Appl*. https://doi.org/10.1007/s00521-020-05655-3
13. Yang T, Qiu W, Ma Y, Chadli M, Zhang L. Fuzzy model-based predictive control of dissolved oxygen in activated sludge processes. *Neurocomputing*. 2014;136(6):88-95.
14. Rahimi H, Kavosi Z, Shojaei P, Kharazmi E. Key performance indicators in hospital based on balanced scorecard model. *J Health Manage Inf*. 2017;4(1):17-24.
15. Shao W, Tian X. Adaptive soft sensor for quality prediction of chemical processes based on selective ensemble of local partial least squares models. *Chem Eng Res Des*. 2015;95:113-132.
16. Sridevi K, Sivaraman E, Mullai P. Back propagation neural network modelling of biodegradation and fermentative biohydrogen production using distillery wastewater in a hybrid upflow anaerobic sludge blanket reactor. *Bioresource Technol*. 2014;165:233-240.
17. Sadeghassadi M, Macnab CJB, Gopaluni B, Westwick D. Application of neural networks for optimal-setpoint design and MPC control in biological wastewater treatment. *Comput Chem Eng*. 2018;115:150-160.
18. Qiao JF, Hou Y, Zhang L, Han HG. Adaptive fuzzy neural network control of wastewater treatment process with multiobjective operation. *Neurocomputing*. 2018;275(1):383-393.
19. Zhou H. Dissolved oxygen control of wastewater treatment process using self-organizing fuzzy neural network. *CIESC J*. 2017;68(4):1516-1524.

20. Han HG, Zhang L, Liu HX, Qiao JF. Multiobjective design of fuzzy neural network controller for wastewater treatment process. *Appl Soft Comput*. 2018;67(3):467-478.
21. Loussifi H, Nouri K, Braiek NB. A new efficient hybrid intelligent method for nonlinear dynamical systems identification: the wavelet kernel fuzzy neural network. *Commun Nonlinear Sci Numer Simulation*. 2016; 32:10-30.
22. Hassen EB, Asmare AM. Predictive performance modeling of Habesha Brewery's wastewater treatment plant using artificial neural networks. *J Env Treat Tech*. 2018;6(2):15-25.
23. Ruan J, Chen X, Huang M, Zhang T. Application of fuzzy neural networks for modeling of biodegradation and biogas production in a full-scale internal circulation anaerobic reactor. *J Env Sci Health Part A*. 2017; 52(1):7-14.
24. Guo Z, Yu K, Jolfaei A, Bashir AK, Almagrabi AO, Kumar N. A fuzzy detection system for rumors through explainable adaptive learning [published online ahead of print January, 2021]. *IEEE Trans Fuzzy Syst*. https://doi.org/10.1109/TFUZZ.2021.3052109
25. Guo Z, Wang H. A deep graph neural network-based mechanism for social recommendations. *IEEE Trans Ind Inf*. 2021;17(4):2776-2783.