Information Sciences 571 (2021) 191–205

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Large-scale water quality prediction with integrated deep neural network ☆

Jing Bi [a], Yongze Lin [a], Quanxi Dong [a], Haitao Yuan [b,*], MengChu Zhou [c]

[a] School of Software Engineering in Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[b] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[c] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

ARTICLE INFO

Article history:
Received 4 November 2020
Received in revised form 27 March 2021
Accepted 11 April 2021
Available online 20 April 2021

Keywords:
Savitzky-Golay filter
Prediction algorithms
Water quality management
Deep neural network
Encoder-decoder network

ABSTRACT

Water environment time series prediction is important to efficient water resource management. Traditional water quality prediction is mainly based on linear models. However, owing to complex conditions of the water environment, there is a lot of noise in the water quality time series, which will seriously affect the accuracy of water quality prediction. In addition, linear models are difficult to deal with the nonlinear relations of data of time series. To address this challenge, this work proposes a hybrid model based on a long short-term memory-based encoder-decoder neural network and a Savitzky-Golay filter. Among them, the filter of Savitzky-Golay can eliminate the potential noise in the time series of water quality, and the long short-term memory can investigate nonlinear characteristics in a complicated water environment. In this way, an integrated model is proposed and effectively obtains statistical characteristics. Realistic data-based experiments prove that its prediction performance is better than its several state-of-the-art peers.

1. Introduction

The serious pollution of a water environment has increasingly attracted people's concerns. Water quality management and its prediction are critically important in a real-life water environment. Dissolved Oxygen (DO) and Chemical Oxygen Demand (CODMn) are important parameters for evaluating water quality. It reflects an equilibrium between the atmosphere exchange rate, and processes of oxygen consumption, *e.g.*, chemical oxidation, nitrification and aerobic respiration, and processes of oxygen production, *e.g.*, photosynthesis. Consequently, it is highly needed to design a water quality prediction method [1].

It is shown that the precise prediction of water quality [2] can help the management of the water environment. Various water quality indicators tend to vary dynamically over time. Accurate water quality prediction can assist the water environment management to make decisions [3], thereby ensuring that water quality values are within a reasonable range. Traditional studies on time series prediction of water quality concentrate on linear methods, *e.g.*, Auto Regression [4], Autoregressive Integrated Moving Average (ARIMA) [12], Moving Average (MA) [5], and other models in statistics [6] and neuron models [7]. Unfortunately, these methods fail to capture nonlinear characteristics in the data.

☆ This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61802015 and 62073005, and in part by the Major Science and Technology Program for Water Pollution Control and Treatment of China under Grant 2018ZX07111005.
* Corresponding author.

https://doi.org/10.1016/j.ins.2021.04.057
0020-0255/

In existing studies, the water quality prediction also adopts traditional support vector machine [13], Artificial Neural Networks (ANNs) [14,15], Deep Belief Networks [8], Stacked AutoEncoder [9], Back Propagation Neural Networks (BPNNs) [10] and other nonlinear methods. Kim et al. [16] select ANN to predict key indicators of time series data of water quality. ANN is an effective model of data-driven prediction that captures nonlinear relations of a time series. Faruk et al. [12] propose a hybrid model that integrates ANN and ARIMA. ANN processes the nonlinear part of a time series and ARIMA processes its linear part. Deep learning has become an emerging research technique from academia to industries [17], and it is widely adopted to predict the time series. Solanki et al. [18] construct a deep learning model by adopting a deep belief network and a denoising autoencoder, and realize the prediction of water quality parameters. Traditional neural networks cannot capture long-term dependencies. Long-term dependencies mean a desired output depends on the inputs presented at times far in the past [45,46]. To solve this challenge, researchers propose a model of Long Short Term Memory (LSTM) [19,20], which adopts memory mechanisms to capture long-term dependencies and is commonly applied in different types of real-life areas [21,22]. Yet, these prediction models often suffer an over-fitting problem due to noise in the data. A Savitzky-Golay (SG) filter [11] can reduce noise interference of time series and increase the prediction accuracy of above models.

This work designs a hybrid model by using an encoder-decoder neural network based on LSTM and a SG filter, named SE-LSTM, to predict the future water quality. We summarize major contributions as follows:

1. This work proposes an improved encoder-decoder network structure, with which the multi-step water quality time series data is better predicted. Thus, the proposed SE-LSTM can better handle long sequences in the time series data.
2. This work innovatively combines and integrates the noise reduction ability of the SG filter, and the feature extraction ability of LSTM to significantly improve the multi-step prediction accuracy.

The remaining sections of this work are given here. Section 2 discusses related studies on the prediction of water quality. Section 3 describes the proposed approach. Section 4 discusses the results of experiments. Finally, Section 5 concludes this work.

2. Related work

Recently, a growing number of studies are proposed to apply time series prediction to water quality. Some existing studies combine traditional water quality prediction algorithms and neural networks to solve the nonlinear problem of prediction of water quality time series. Noori et al. [42] adopt a hybrid model by combining a SWAT and an ANN. They optimize the water quality prediction model by using complex water quality processes and hydrological mechanisms. Khan et al. [39] combine an ANN and an Adaptive Neuro-Fuzzy Inference System (ANFIS) to build an ensemble model. The yielded ANN-ANFIS model well describes nonlinear input–output relations of complex datasets. However, it does not consider the multi-step prediction. Babu et al. [5] propose a new prediction model based on ARIMA-ANN that applies nonlinear ANN and linear ARIMA for higher accuracy of prediction. This model can be applied to the one-step and multi-step prediction. Yet the linear characteristic in the hybrid model affects the accuracy of multi-step prediction. Li et al. [41] design a hybrid prediction model by combing LSTM and sparse auto-encoder. The data in a hidden layer pre-trained by the sparse auto-encoder includes latent features of quality of water. This deep model effectively improves the accuracy of multi-step prediction. Dong et al. [47] adopt a basic LSTM model to predict the water quality. Bi et al. [48] adopt an attention mechanism to predict the water quality. However, these models fail to improve the network structure of LSTM. Different from above models, our proposed model improves the encoder-decoder network structure so that it can better adapt to the multi-step time series prediction. At the same time, to solve the noise reduction problem of the time series data, this work adopts an SG filter to denoise the original data.

There are also some studies that use filtering to denoise time series to improve the accuracy of prediction models. Lu et al. [43] propose decision tree-based models to predict the water quality, and an advanced denoising method to preprocess raw data. The model achieves better accuracy in short-term forecasting of the water quality data. Xu et al. [40] combine BPNNs and wavelet transform to build a prediction model for water quality. It achieves a high training speed and strong robustness. Najah et al. [44] implement enhanced wavelet de-noising techniques, and adopt a prediction model based on them to effectively improve the accuracy of the water quality prediction. They effectively reduce the impact of noise in the time series on the prediction accuracy. However, these models cannot effectively capture the long-term features of time series data. Different from these studies, the SG filter used in our model can effectively retain the characteristics of a time series while removing its noise. At the same time, combined with an encoder-decoder model based on LSTM, our model significantly improves the accuracy of multiple-step prediction.

3. Model architecture

We further present a hybrid architecture that is derived by combing the SG filter and an encoder-decoder network based on LSTM to achieve the prediction of the future water quality.

3.1. Problem formulation

$X = \{x_1, \ldots, x_t, \ldots, x_T\}$ denotes the time series of water quality, and T represents previous T time steps. $\bar{x}$ represents the value of water quality processed by the SG filter. $\hat{y}$ denotes a predicted sequence of water quality values. $\widehat{Y} = \{\hat{y}_1, \ldots, \hat{y}_{t'}, \ldots, \hat{y}_\tau\}$ represents the predicted value in the next τ time steps.

3.2. SG filter

To eliminate frequent variations in water quality, we use the filter of SG [11] to decrease the interference of noise. The filter retains the effective information of data in the denoising process. Then, each sub-sequence of data is fitted by the linear least squares approach. This method is called convolution [23]. The polynomial is obtained as:

$$\bar{x}_\zeta = \sum_{r=\frac{1-m}{2}}^{\frac{m-1}{2}} c_r x_{r+\zeta}, \qquad \frac{m-1}{2} \leqslant \zeta \leqslant t - \frac{m-1}{2} \tag{1}$$

where m is the filter window width and $x_{r+\zeta}$ is a data point in the filter window. The convolution coefficient $C_\zeta = \left\{c_{\frac{1-m}{2}}, \ldots, c_r, \ldots, c_{\frac{m-1}{2}}\right\}$ gives prediction of the smoothed data $x_{r+\zeta}$ in each filter window. $\bar{x}_\zeta$ represents the filtered results.

The window's $x_{r+\zeta}$ $(\zeta = 1, \ldots, T)$ is estimated with a method of least-squares that has m coefficients of convolution. The window size will affect the results with the SG filter.

Calculation of coefficients of convolution C_ζ: It is fit to adjacent data points $X = \left\{x_{\frac{1-m}{2}}, \ldots, x_0, \ldots, x_{\frac{m-1}{2}}\right\}$ by the method of linear least squares. The width of the filter window is denoted by m. $z = \{\frac{1-m}{2}, \ldots, 0, \ldots, \frac{m-1}{2}\}$, and z is an integer. The coefficients including a_0, a_1, *etc. are obtained by using the linear least squares approach.* x_r *in the window is obtained as follows.*

$$x_r = a_0 + a_1 z_r + a_2 z_r^2 + \cdots + a_{k-1} z_r^{k-1} \tag{2}$$

There are m equations of x_r, and a matrix X expressed is given as:

$$X = Z \cdot A \tag{3}$$

where $X \in \mathbb{R}^{m\times 1}, Z \in \mathbb{R}^{m\times k}$ and $A \in \mathbb{R}^{k\times 1}$. A is determined by the method of least-squares fitting.

Then, $\widehat{A}$ is calculated by the method of least squares:

$$\widehat{A} = \left(Z^{\mathrm{T}} \cdot Z\right)^{-1} \cdot Z^{\mathrm{T}} \cdot X \tag{4}$$

Finally, the matrix C_ζ of convolution coefficients of X is obtained as:

$$C_\zeta = \left(Z^{\mathrm{T}} \cdot Z\right)^{-1} \cdot Z^{\mathrm{T}} \tag{5}$$

3.3. Proposed method (SE-LSTM)

After the reduction of noise, this work adopts an LSTM [19] model to predict the quality of water. LSTM can learn from important experiences, which have long-term relations of dependencies. The memory cells of LSTM are illustrated in Fig. 1.

In Fig. 1, this work focuses on the analysis of memory cells of LSTM at time step $t \in T$. LSTM memory cell consists of three gates, which are **forget**, **input** and **output** ones [24], σ denotes the activation function. In addition, c_t denotes the state of a memory cell, h_t denotes the output of an LSTM unit, and there are three gates $(f_t; i_t; o_t)$ executing functions of sigmoid to

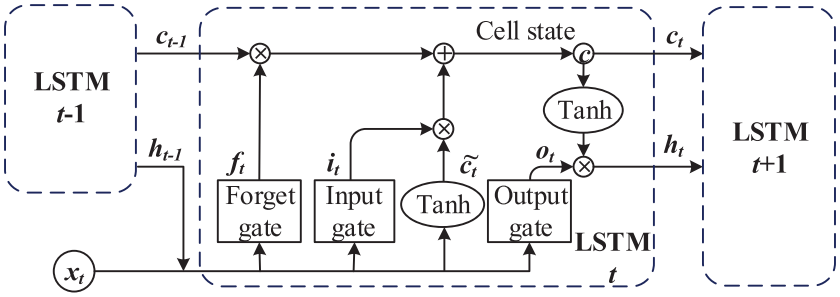


Fig. 1. LSTM units.

update the memory at t. f_t, i_t and o_t are the calculation approaches for the gates of **input**, **output** and **forget** at t, respectively. Thus,

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{6}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{7}$$

$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{9}$$

In the output gate, the sigmoid function specifies the dropped part of a state of cell. The tanh function deals with the current cell state c_t, and it is further multiplied with the result of the sigmoid function. Then, the following output at t, o_t, is produced.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{10}$$

$$h_t = o_t \odot tanh(c_t) \tag{11}$$

where $\odot$ denotes the operation of dot-wise product, the matrices W_i, W_f, and W_o denote the parameters of the **input**, **forget** and **output** gates, respectively. W_c denotes parameters of the memory cell. $tanh(\cdot)$ and $\sigma(\cdot)$ denotes functions of hyperbolic and sigmoid, respectively.

The back propagation (BP) algorithm [25] of LSTM [26] is consistent with that of RNN.

To achieve better prediction accuracy, this work proposes a novel Encoder-Decoder neural network, which is illustrated in Fig. 2.

The encoder network in Fig. 2 is an LSTM that sequentially transforms the input $\{x_1, \ldots, x_t, \ldots, x_T\}$. Then, a map from x_t to h_t at t is obtained with

$$h_t = f_1(h_{t-1}, x_t) \tag{12}$$

where h_t is the encoder's hidden state at t, and f_1 is an unit of LSTM.

c_T is the hidden state information obtained by the LSTM-based encoder. The decoder is also an LSTM that produces the output by obtaining $\{\hat{y}_1, \ldots, \hat{y}_{t'}, \ldots, \hat{y}_\tau\}$ given the hidden state $d_{t'}$, while $d_{t'}$ is conditioned on c_T. Let $d_{t'}$ denote the hidden state information of the decoder at t'. Thus, $d_{t'}$ is obtained as:

$$d_{t'} = f_2(d_{t'-1}, c_T) \tag{13}$$

f_2 is another LSTM unit. LSTM extracts complicated characteristics of a time series. In the prediction of multiple steps, the output of the encoder, c_T, and the last step of the hidden state $d_{t'-1}$ from the decoder can improve the performance of the multi-step prediction. A fully connected layer can output them into the predicted values as follows.

$$\hat{y}_{t'} = w_k^T \cdot d_{t'} \tag{14}$$

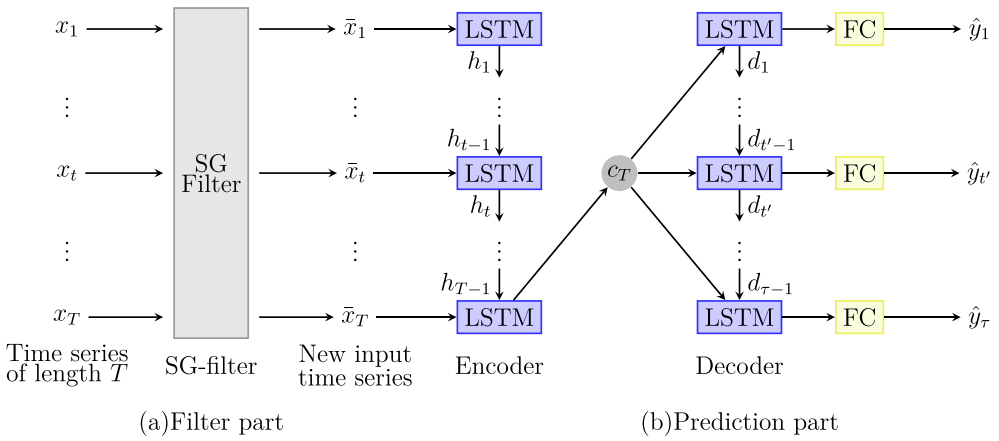


Fig. 2. SE-LSTM model network structure. (a) The SG filter is adopted to reduce the input noise in $\{x_1, \ldots, x_t, \ldots, x_T\}$. (b) After noise reduction, the prediction mechanism encodes $\{\bar{x}_1, \ldots, \bar{x}_t, \ldots, \bar{x}_T\}$. The encoder output c_T and the hidden state $d_{t'-1}$ of decoder previous step are adopted as input for decoding of current step t'. The output of decoder is sent to a final layer that is fully connected to obtain $\{\hat{y}_1, \ldots, \hat{y}_{t'}, \ldots, \hat{y}_\tau\}$.

where w is the weight parameter in the final layer that is fully connected, k denotes the layer number, and $\hat{y}_{t'}$ denotes the predicted value at t'. The details for the training process of SE-LSTM are given in Algorithm 1.

Algorithm 1. Training process of SE-LSTM

Input: Water quality sequences: $\{x_1, x_2, \ldots, x_T\}$
Output: Predicted sequences: $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_\tau\}$
1: Initialize window size (m), polynomial degree (k), learning rate: (lr), number of input steps (T), number of prediction steps (τ), hidden states ($p = q$).
2: **for** $\zeta = 1$ to T **do**
3: Obtain the smoothed time series $\bar{x}_\zeta$ with (1);
4: **end for**
5: **for** each epoch **do**
6: **for** $t = 1$ to T **do**
7: Generate h_t and c_t *via* encoder of LSTM in (6)–(9) with the inputted $\bar{x}_t$ and h_{t-1};
8: **end for**
9: **for** $t' = 1$ to τ **do**
10: Generate $d_{t'}$ *via* decoder of LSTM in (6)–(9) with the inputted c_T and $d_{t'-1}$;
11: Generate $\hat{y}_{t'}$ with (14);
12: **end for**
13: Compute loss $\mathscr{L}$ with (15);
14: Apply BPTT to backpropagate the gradient;
15: Train the model for minimizing the loss with the Adam optimizer;
16: Update lr
17: **end for**

3.4. Parameter learning

Here, MAE [27] is adopted as the function of loss ($\mathscr{L}$) of SE-LSTM. $\mathscr{L}$ is calculated as follows.

$$\mathscr{L} = \frac{1}{\tau}\sum_{t'=1}^{\tau}\left|Y - \widehat{Y}\right| \tag{15}$$

where τ is the number of prediction values. $\widehat{Y}$ and Y are the predicted value of the proposed method and the real one of water quality at t', respectively.

SE-LSTM adopts the gradient reduction algorithm in the Adam optimizer [28]. Here, to effectively train the model, the initial rate of learning is set to 0.01, and we calculate the training loss by the decay of 1×10^{-6}.

4. Performance evaluation

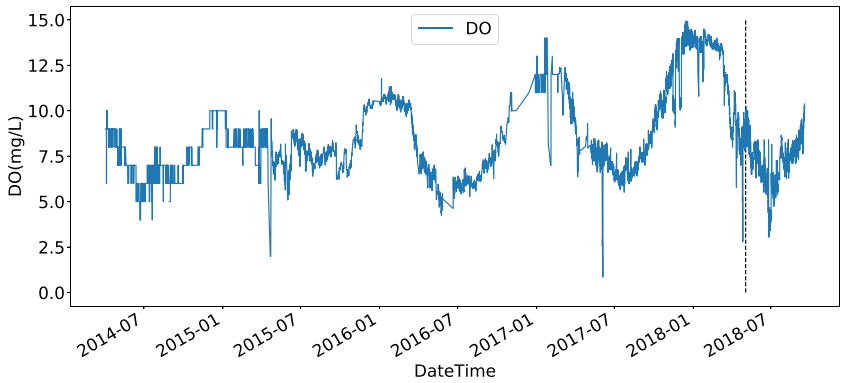
4.1. Experimental setup

The SE-LSTM and other baseline models are all implemented in PYTORCH and tested in a system with Intel Xeon CPU E5-2683 and GTX1080 GPU to evaluate their performance. We test different baseline methods and determine their optimal parameter settings. The experimental dataset of the water environment[1] is collected from GuBeiKou, Beijing, China. The data is collected every four hours, with a total of more than 10,000 pieces of data from April 2014 to October 2018. The dataset is then split into two parts including the training and the test sets with a ratio of 9:1. DO and CODMn are adopted as experimental data for all prediction models. The distributions of DO and CODMn are visualized in Fig. 3.

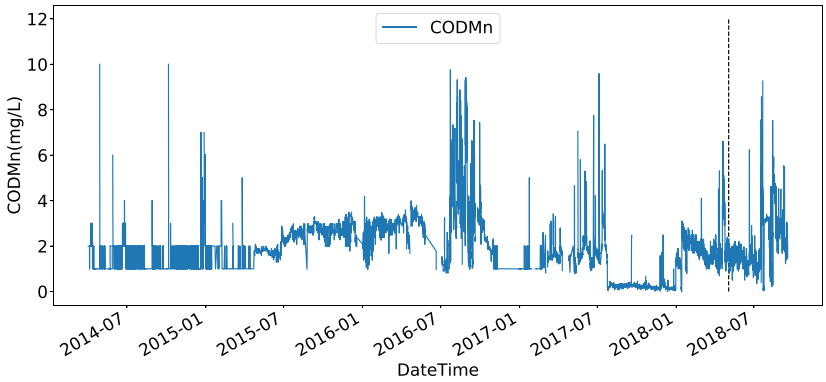
In the data preprocessing, this work adopts a smoothing operation to eliminate instability factors, but it also removes some effective information in the original time series. Thus, several experiments are implemented to demonstrate the prediction accuracy of the proposed method. First, we present the best filter through different window sizes in our comparative experiments. In Moving Median (MM) [29], SG and MA filters of the original time series, we need to first specify the window size in the filter operation. As illustrated in Table 1, we obtain the smallest values of M_{SE} in three filters when the window size is 5.

Then, we further choose the first window data from the original time series, and the average or median value as the fit value in the current window. Then, the window is slided until all the data points in the original time series are replaced with fit values. The first and last values in time series are removed. To keep the length of time series processed by the filter oper-

[1] http://data.epmap.org/water.



(a) Data distribution of DO.



(b) Data distribution of CODMn.

Fig. 3. Distribution of GuBeiKou water quality data.

Table 1
RMSE with different window sizes.

	RMSE		
Window size	MM Filter	MA Filter	SG Filter
3	0.48	0.62	0.57
5	0.61	0.69	**0.32**
7	0.57	0.68	0.39
11	0.62	0.62	0.37
15	0.63	0.63	0.42

ation, the smoothed time series adds these two values removed from the original time series at the beginning and end, respectively.

Furthermore, we give different parameter settings for obtaining better SG filter, and the results are illustrated in Table 2. The smaller Root Mean Square Error (RMSE) [30] is obtained when the window size is 5, *i.e.*, $m = 5$, and the polynomial of degree k is set to 3. Too large window size removes the time features, while too small window size cannot be used to reduce noise. In the same way, too large k leads to the overfitting of least squares, too small k causes an underfitting issue.

4.2. Evaluation comparison

To prove the prediction performance of SE-LSTM, three metrics are adopted to evaluate its prediction accuracy. Specifically, MAE [27], Mean Absolute Percentage Error (MAPE) [31] and RMSE are adopted.

Table 2
Different parameter settings for the SG filter.

Window Size (m)	Polynomial Degree (k)	RMSE
3	3	0.57
5	**3**	**0.32**
11	5	0.37
7	3	0.39
11	7	0.41
15	7	0.42
15	9	0.43
11	9	0.45
11	3	0.49

Table 3
Comparison results over DO dataset.

Methods	Steps														
	Step 5			Step 4			Step 3			Step 2			Step 1		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SVR	0.492	0.754	7.66	0.482	0.739	7.48	0.468	0.719	7.23	0.442	0.683	6.79	0.405	0.623	6.17
XGBoost	0.492	0.747	7.66	0.489	0.729	7.51	0.477	0.707	7.27	0.453	0.671	6.85	0.405	0.623	6.26
ARIMA	0.475	0.734	6.96	0.456	0.707	6.73	0.448	0.685	6.55	0.552	0.893	8.11	0.38	0.582	5.55
LSTM	0.492	0.754	8.16	0.546	0.831	8.38	0.559	0.844	8.21	0.459	0.709	6.98	0.39	0.595	5.85
ANN	0.581	0.799	8.88	0.593	0.803	8.381	0.574	0.806	8.21	0.55	0.747	8.11	0.514	0.681	7.658
Seq2seq	0.472	0.729	7.15	0.519	0.726	7.57	0.44	0.676	6.47	0.42	0.648	6.36	0.37	0.554	5.59
SG-SVR	0.453	0.692	6.95	0.434	0.668	7.57	0.404	0.629	6.18	0.354	0.558	32.27	0.25	0.398	3.87
SG-XGBoost	0.473	0.701	7.26	0.461	0.68	7.04	0.44	0.653	6.71	0.411	0.601	6.17	0.34	0.483	5.09
SG-ARIMA	0.465	0.694	6.86	0.442	0.668	6.51	0.401	0.613	5.87	0.353	0.554	5.11	**0.201**	**0.292**	**2.90**
SG-LSTM	0.461	0.696	7.07	0.441	0.652	6.64	0.389	0.594	5.78	0.331	0.517	4.79	0.22	0.321	5.59
SG-ANN	0.444	0.678	6.65	0.454	0.679	6.76	0.399	0.617	5.78	0.371	0.548	5.53	0.31	0.424	4.56
SE-LSTM	**0.431**	**0.641**	**6.48**	**0.405**	**0.625**	**6.14**	**0.376**	**0.588**	**5.51**	**0.319**	**0.509**	**4.71**	**0.204**	**0.309**	**2.99**

Table 4
Comparison results over CODMn dataset

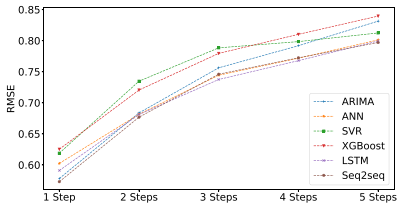
Methods	Steps														
	Step 5			Step 4			Step 3			Step 2			Step 1		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SVR	0.558	0.955	86.38	0.551	0.944	86.17	0.543	0.932	85.857	0.531	0.913	85.82	0.516	0.873	53.11
XGBoost	0.535	0.895	61.27	0.523	0.877	58.14	0.515	0.866	55.26	0.47	0.799	44.08	0.47	0.799	44.08
ARIMA	0.525	0.869	59.9	0.516	0.852	57.08	0.526	0.873	59.24	0.585	0.988	51.98	0.463	0.773	44.51
LSTM	0.544	0.925	56.96	0.52	0.875	52.92	0.498	0.843	51.63	0.476	0.811	46.4	0.468	0.784	49.35
ANN	0.593	0.997	63.06	0.596	0.999	61.38	0.557	0.949	56.82	0.587	0.9727	67.39	0.516	0.873	53.11
Seq2seq	0.512	0.856	49.59	0.501	0.841	45.98	0.488	0.82	42.04	0.477	0.789	42.26	0.467	0.773	48.74
SG-SVR	0.535	0.922	81.77	0.522	0.902	81.24	0.503	0.874	80.34	0.469	0.825	78.72	0.383	0.691	73.89
SG-XGBoost	0.505	0.839	57.1	0.488	0.811	53.5	0.466	0.777	49.15	0.427	0.718	41.57	0.354	0.576	32.26
SG-ARIMA	0.479	0.818	53.89	0.465	0.802	51.71	0.438	0.759	45.03	0.414	0.745	35.03	**0.249**	**0.429**	**21.66**
SG-LSTM	0.503	0.854	59.85	0.478	0.817	51.11	0.441	0.764	45.56	0.393	0.679	40.05	0.278	0.454	24.16
SG-ANN	0.516	0.868	50.13	0.497	0.837	51.86	0.47	0.787	48.3	0.416	0.719	39.33	0.308	0.505	28.52
SE-LSTM	**0.473**	**0.814**	**45.83**	**0.455**	**0.784**	**49.77**	**0.428**	**0.738**	**39.18**	**0.391**	**0.679**	**36.49**	**0.265**	**0.441**	**21.81**

$$\mathrm{MAE} = \frac{1}{\tau}\sum_{t'=1}^{\tau}|y_{t'} - \hat{y}_{t'}| \tag{16}$$

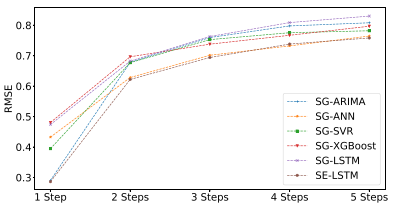
$$\mathrm{RMSE} = \sqrt{\frac{1}{\tau}\sum_{t'=1}^{\tau}(y_{t'} - \hat{y}_{t'})^2} \tag{17}$$

$$\mathrm{MAPE} = \frac{100}{\tau}\sum_{t'=1}^{\tau}\left|\frac{y_{t'} - \hat{y}_{t'}}{y_{t'}}\right| \tag{18}$$

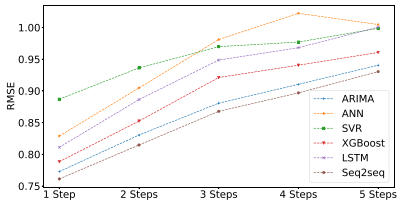
where τ is the sequence length, $y_{t'}$ and $\hat{y}_{t'}$ are real and predicted values of water quality at t', respectively.



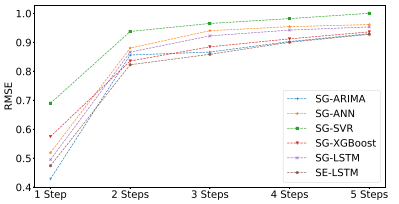
(a) RMSE with increasing periods over DO.



(b) RMSE with increasing periods with the filter of SG over DO.

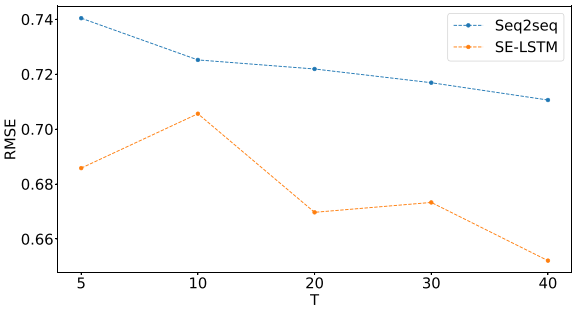


(c) RMSE with increasing periods with the filter of SG over CODMn.

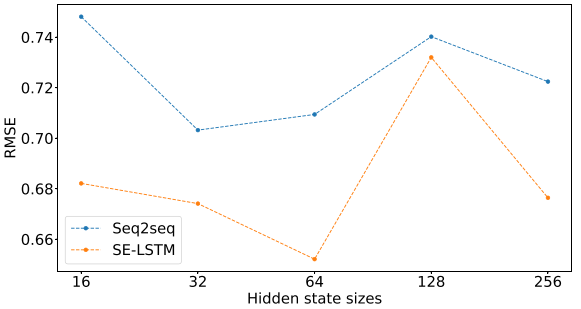


(d) RMSE with increasing periods with the filter of SG over CODMn.

Fig. 4. RMSE of each method with and without the filter of SG.



(a) RMSE vs. T.

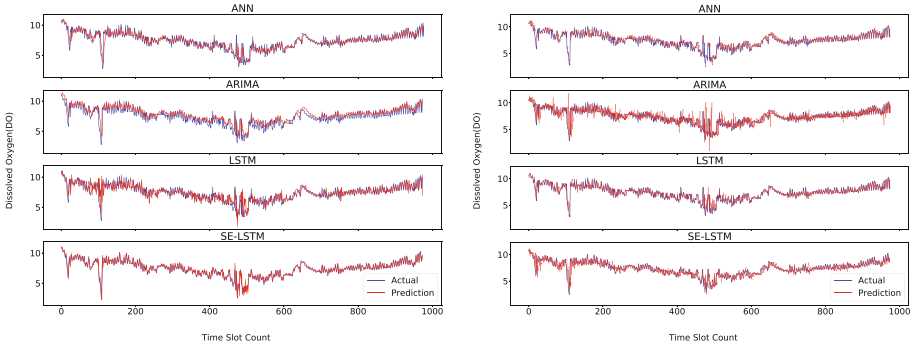


(b) RMSE vs. Hidden state sizes.

Fig. 5. Performance among different variants. Seq2seq mainly removes the SG filter based on the structure of SE-LSTM.

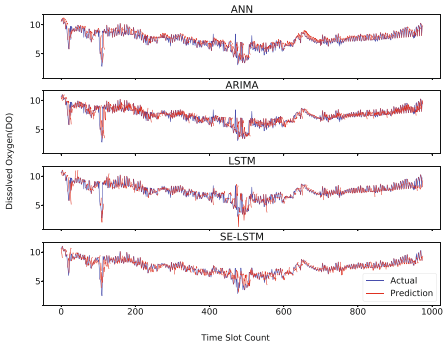
4.3. Hyperparameter setting

In the training phase, $\tau \in \{1, 2, 3, 4, 5\}$ and the batch size is 64. The learning rate is 0.01, *i.e.*, $lr = 0.01$. In the SG filter, $m \in \{3, 5, 7, 11, 15\}$ and $k \in \{3, 5, 7, 9\}$. In the encoder-decoder module, $T \in \{5, 10, 20, 30, 40\}$. In addition, $p = q \in \{16, 32, 64, 128, 256\}$, and the best performance is achieved when $T = 40$ and $p = q = 64$ in the validation set. We have two main ways to determine the parameters of the neural network. First, we refer to the setting of parameters given in existing similar studies to determine the parameters of our model [49]. For example, batch size and learning rate are set in this way. Second, we adopt comparative experiments to find the best parameter settings. For example, the hidden size and the input time steps are set in this way.

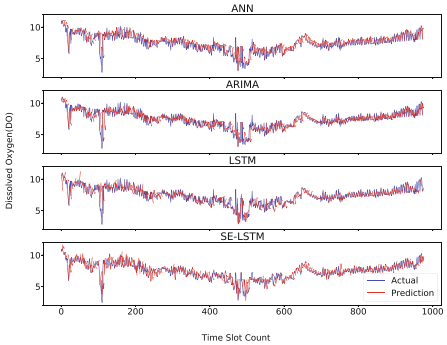


(a) Step 1

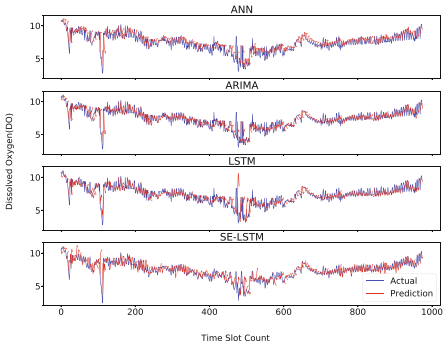
(b) Step 2



(c) Step 3



(d) Step 4



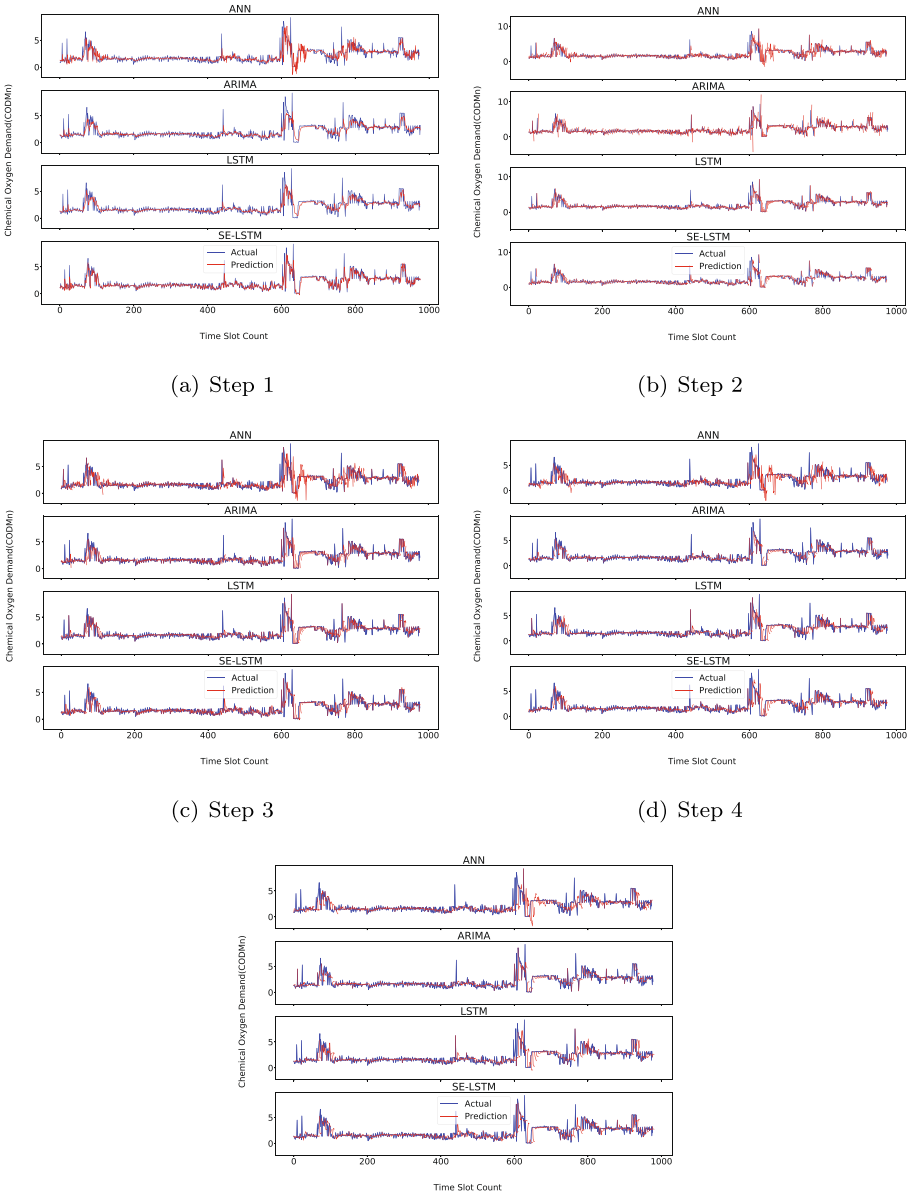
(e) Step 5

Fig. 6. Predicted curves from step 1 to 5 over DO.

4.4. Benchmarks

To compare the proposed method from different aspects, we select several benchmark models for comparison.

1. **ARIMA.** In ARIMA, past values and random noise are fitted to the predicted value through a linear function.
2. **ANN.** ANN can well handle nonlinear characteristics, and has good prediction ability.
3. **eXtreme Gradient Boosting (XGBoost).** XGBoost [32,33] is a general Tree Boosting algorithm. It is widely adopted in machine learning areas. It first constructs several trees of regression with tree ensemble techniques, and the total scores of all trees of regression are used as the finally predicted value.
4. **SVR.** SVR [38,34] can improve the ability of generalization by obtaining the least structural risk. In the case of less statistical samples, it also obtains good statistical rules.



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

(e) Step 5

Fig. 7. Predicted curves from step 1 to 5 over CODMn.

5. **LSTM.** LSTM is a commonly applied sequence prediction model [35,36]. Particularly, LSTM is excellent in producing long-term dependency of time series. To show the performance comparison of LSTM and SE-LSTM, their settings of hyperparameters are the same in the experiment.

4.5. Quantitative comparison

Then, SE-LSTM is further compared with its typical benchmark methods in this experiment. These methods include linear models, *e.g.*, ARIMA, traditional nonlinear models, *e.g.*, ANN [37], SVR, XGBoost, and deep models, *e.g.*, LSTM. MAPE, MAE and RMSE of SE-LSTM and other baseline ones from step 1 to 5 are given in Tables 3 and 4. The experimental results show that traditional nonlinear models do not have much advantage over linear models. This is because the water quality time series has stronger linear characteristics, and linear models have certain advantages. In addition, the prediction accuracy of deep learning is better than traditional models, and SE-LSTM is better than SG-LSTM in terms of the multi-step prediction accuracy due to changes in the network structure. Besides, after adding the SG filter, the prediction accuracy of all models has been significantly improved, and it validates that the SG filter can better retain the time series characteristics and effectively remove noise. Although the ARIMA model is slightly ahead of the SE-LSTM model in a single step, SE-LSTM performs better than ARIMA in the multi-step prediction. It is observed that SE-LSTM outperforms its peers with respect to three metrics.

4.6. Variant comparison

Fig. 4 gives the predicted result in each step with all methods over DO and CODMn. RMSE of all models increases with the increase of the step size. Note that RMSE of the SE-LSTM model with the first five steps is relatively low, while baseline methods have consistently higher RMSE. Compared with the filtered and unfiltered baseline methods, Seq2seq and SE-LSTM obtain the best performance. Therefore, the multi-step prediction effect of SE-LSTM is better than other methods.

We further evaluate the SE-LSTM's sensitivity in terms of its parameters. Other parameters keep unchanged when T or p (q) is changed. Seq2seq mainly removes the SG-filter based on the structure of SE-LSTM. Fig. 5(a) shows the RMSE with different time steps. It is observed that SE-LSTM achieves the best results when $T = 40$. Fig. 5(b) illustrates the RMSE with dif-

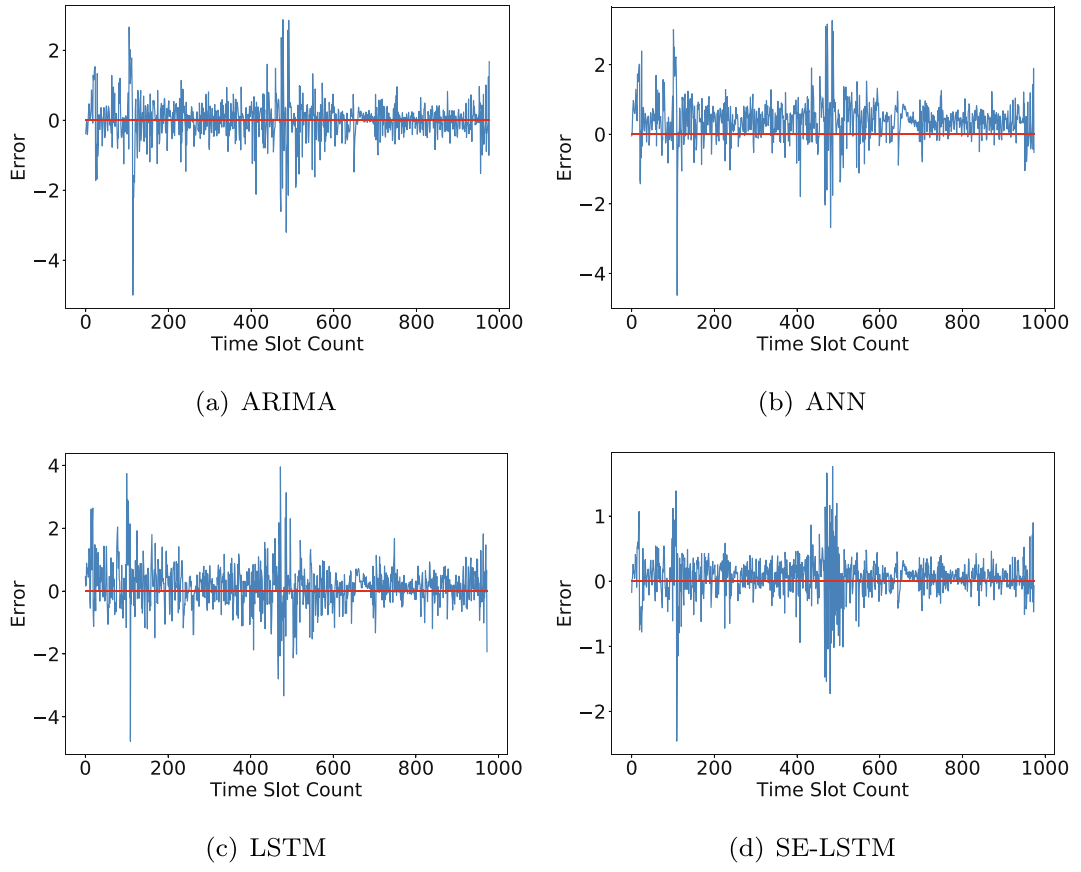


Fig. 8. Prediction errors over DO.

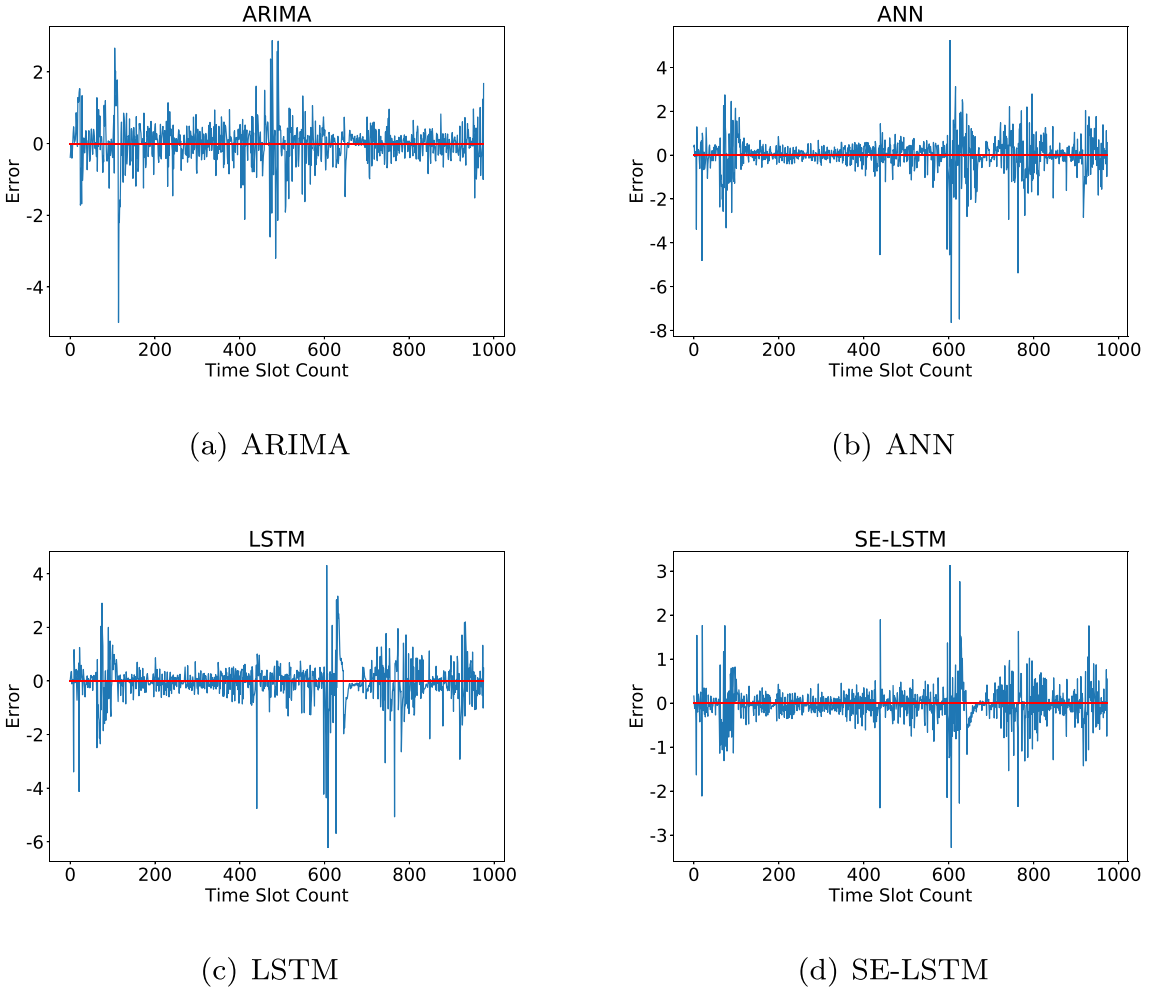


Fig. 9. Prediction errors over CODMn.

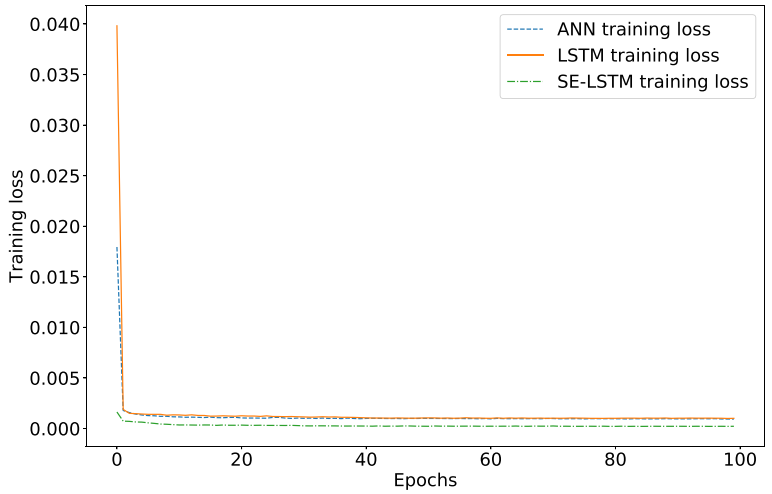


Fig. 10. Training loss of SE-LSTM and other baseline methods over DO.

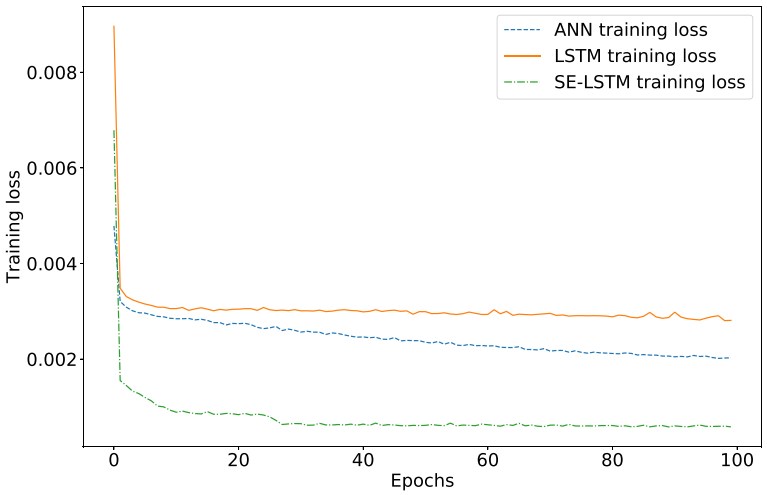


Fig. 11. Training loss of SE-LSTM and other baseline methods over CODMn.

ferent hidden state sizes for the encoder and decoder LSTM networks ($p = q \in \{16, 32, 64, 128, 256\}$). It is shown that the optimal result is obtained when $p = q = 64$.

To better evaluate the multi-step prediction of the model, this experiment changes the prediction step size from 1 to 5. Figs. 6 and 7 show the predicted fitting curves of four models, the blue and red curves mean the real and predicted values. It is observed that LSTM and SE-LSTM outperform others in the multi-step fitting, which is because of feature extraction function of LSTM. In addition, the RMSE of SE-LSTM is 15.6% lower than that of LSTM due to the effect of the SG filter operation. Thus, SE-LSTM achieves the best prediction result in the experiment.

Figs. 8 and 9 show that the error ranges of ARIMA, ANN and LSTM vary around -2 to 2, while that of SE-LSTM varies from -1 to 1. Thus, SE-LSTM achieves a smaller error range than other methods. Figs. 10 and 11 show the training loss in ANN, LSTM, and SE-LSTM. After 200 epochs, ANN, LSTM and SE-LSTM reach 0.03, 0.02 and 0.001, respectively. Thus, the training convergence efficiency of SE-LSTM is better than other models.

5. Conclusion

Water environment management needs an accurate water quality prediction model. It is a big challenge to precisely forecast it because of its highly complex and varying characteristics. This work proposes a hybrid prediction model, which investigates an encoder-decoder neural network with long short-term memory and a Savitzky-Golay filter. The resulting prediction model is achieved to forecast the time series of water quality in the future. We apply a filter of Savitzky-Golay to eliminate the noise in the time series data. This work adopts an encoder-decoder model based on LSTM to capture features. Finally, the experimental results with real-life data prove that the proposed model provides more accurate prediction than several benchmark models. Our future work will further extend the current model from the following two points: 1) We will study related dimensionality reduction algorithms, and reducing the interference of correlated noise by using some smoothing methods to better extract temporal characteristics; and 2) We will further improve the network structure of current model, and therefore, it can not only receive historical data, but also capture the input of related multi-feature data; 3) We plan to apply our current model to other fields, *e.g.*, financial time series [50], and traffic flow, to verify its effectiveness and robustness; 4) We plan to combine graph neural networks and time series models to extract temporal and spatial characteristics of the water quality; and 5) we plan to theoretically investigate and discuss the convergence of filtering and prediction models.

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.

CRediT authorship contribution statement

Jing Bi: Conceptualization, Methodology, Funding acquisition, Writing - original draft, Project administration. **Yongze Lin:** Writing - original draft. **Quanxi Dong:** Visualization. **Haitao Yuan:** Formal analysis, Investigation, Validation, Supervision. **MengChu Zhou:** Writing - review & editing.

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.

References

[1] S. Palani, S.Y. Liong, P. Tkalich, J. Palanichamy, Development of a neural network model for dissolved oxygen in seawater, Approximation with Artificial Neural Networks 38 (2) (2009) 27–38.
[2] K. George, P. Mutalik, A multiple model approach to time-series prediction using an online sequential learning algorithm, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (5) (2019) 976–990.
[3] W. Zhao, T.H. Beach, Y. Rezgui, Automated model construction for combined sewer overflow prediction based on efficient LASSO algorithm, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (6) (2019) 1254–1269.
[4] Q. Cai, D. Zhang, W. Zheng, S.C. Leung, A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression, Knowledge-Based Systems 74 (2015) 61–68.
[5] C.N. Babu, B.E. Reddy, A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data, Applied Soft Computing 23 (2014) 27–38.
[6] E.S. Gardner Jr, Exponential smoothing: the state of the art-Part II, International Journal of Forecasting 22 (4) (2006) 637–666.
[7] S Gao et al, Dendritic neuron model with effective learning algorithms for classification, approximation and prediction, IEEE Transactions on Neural Networks and Learning Systems 30 (2) (2019) 601–614.
[8] G.E. Hinton, Deep belief networks, Scholarpedia 4 (5) (2009) 5947.
[9] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2015) 865–873.
[10] J. Qiao et al, A self-organizing RBF neural network based on distance concentration immune algorithm, IEEE/CAA Journal of Automatica Sinica 7 (1) (2020) 276–291, https://doi.org/10.1109/JAS.2019.1911852.
[11] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Analytical Chemistry 36 (1964) 1627–1639.
[12] D.Ömer Faruk, A hybrid neural network and ARIMA model for water quality time series prediction, Engineering Applications of Artificial Intelligence 23 (4) (2010) 586–594..
[13] G. Wang, G. Zhang, K. Choi, J. Lu, Deep additive least squares support vector machines for classification with model transfer, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (7) (2019) 1527–1540.
[14] C.F. Stallmann, A.P. Engelbrecht, Gramophone noise detection and reconstruction using time delay artificial neural networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (6) (2017) 893–905.
[15] D.M. Sahoo, S. Chakraverty, Functional link neural network learning for response prediction of tall shear buildings with respect to earthquake data, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (1) (2018) 1–10.
[16] S.E. Kim, I.W. Seo, Artificial neural network ensemble modeling with conjunctive data clustering for water quality prediction in rivers, Journal of Hydro-environment Research 9 (3) (Sept. 2015) 325–339.
[17] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Computation 18 (7) (2006) 1527–1554.
[18] A. Solanki, H. Agrawal, K. Khare, Predictive analysis of water quality parameters using deep learning, International Journal of Computer Applications 125 (9) (2015) 0975–8887.
[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
[20] Z. Ren, K. Qian, Z.X. Zhang, V. Pandit, A. Baird, B. Schuller, Deep scalogram representations for acoustic scene classification, IEEE/CAA Journal of Automatica Sinica 5 (3) (2018) 662–669.
[21] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space Odyssey, IEEE Transactions on Neural Networks and Learning Systems 703 (10) (2017) 2222–2232.
[22] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transportation Research Part C: Emerging Technologies 54 (2015) 187–197.
[23] Y. Pang, M. Sun, X. Jiang, X. Li, Convolution in convolution for network in network, IEEE Transactions on Neural Networks and Learning Systems 29 (5) (2018) 1587–1597.
[24] D. Tao, X. Lin, L. Jin, X. Li, Principal component 2-D long short-term memory for font recognition on single chinese characters, IEEE Transactions on Cybernetics 46 (3) (2016) 756–765.
[25] R. Koda, T. Origasa, T. Nakajima, Y. Yamakoshi, Observing bubble cavitation by back-propagation of acoustic emission signals, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 66 (5) (2019) 823–833.
[26] H. Liu et al, Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods, IEEE Transactions on Computational Social Systems 7 (6) (2020) 1358–1375, https://doi.org/10.1109/TCSS.2020.3033302.
[27] C.J. Willmott, K. Matsuura, Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in assessing average model performance, Climate Research 30 (1) (2005) 79–82.
[28] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: Proc. 3rd International Conf. for Learning Representations, San Diego, CA, USA, 2015, pp. 1–15..
[29] A.F. Bastos, K. Lao, G. Todeschini, S. Santoso, Novel moving average filter for detecting RMS voltage step changes in triggerless PQ data, IEEE Transactions on Power Delivery 33 (6) (2018) 2920–2929.
[30] T. Chai, R.R. Draxler, Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?-Arguments against avoiding RMSE in the literature, Geoscientific Model Development 7 (3) (2014) 1247–1250.
[31] M. Jorgensen, Experience with the accuracy of software maintenance task effort prediction models, IEEE Transactions on Software Engineering 21 (8) (1995) 674–681.
[32] J. Zhong, Y. Sun, W. Peng, M. Xie, J. Yang, X. Tang, XGBFEMF: An XGBoost-based framework for essential protein prediction, IEEE Transactions on NanoBioscience 17 (3) (Jul. 2018) 243–250.
[33] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proc. of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2016, pp. 785–794..
[34] A. Melingui, J.J.M. Ahanda, O. Lakhal, J.B. Mbede, R. Merzouki, Adaptive algorithms for performance improvement of a class of continuum manipulators, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (9) (2018) 1531–1541.
[35] L. Jin, S. Li, B. Hu, RNN models for dynamic matrix inversion: a control-theoretical perspective, IEEE Transactions on Industrial Informatics 14 (1) (2018) 189–199.
[36] Z. Zhang, L. Kong, L. Zheng, Power-type varying-parameter RNN for solving TVQP problems: design, analysis, and applications, IEEE Transactions on Neural Networks and Learning Systems 30 (8) (2019) 2419–2433.
[37] R. Kamesh, K.Y. Rani, Novel formulation of adaptive MPC as EKF using ANN model: multiproduct semibatch polymerization reactor case study, IEEE Transactions on Neural Networks and Learning Systems 28 (12) (2017) 3061–3073.
[38] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (3) (2004) 199–222.

[39] Y. Khan, S.S. Chai, Ensemble of ANN and ANFIS for water quality prediction and analysis - a data driven approach, Journal of Telecommunication Electronic and Computer Engineering (JTEC) 9 (2–9) (2017) 117–122.
[40] L. Xu, S. Liu, Study of short-term water quality prediction model based on wavelet neural network, Mathematical and Computer Modelling 58 (3) (2013) 807–813.
[41] Z. Li, F. Peng, B. Niu, G. Li, J. Wu, Z. Miao, Water quality prediction model combining sparse auto-encoder and LSTM network, IFAC-Papers OnLine 51 (17) (2018) 831–836.
[42] N. Noori, L. Kalin, S. Isik, Water quality prediction using SWAT-ANN coupled approach, Journal of Hydrology 590 (2020) 125220.
[43] H. Lu, X. Ma, Hybrid decision tree-based machine learning models for short-term water quality prediction, Chemosphere 249 (2020) 126169.
[44] 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, Journal of Hydrology 578 (2019) 124084.
[45] R. Li, Y. Huang, J. Wang, Long-term traffic volume prediction based on K-means Gaussian interval type-2 fuzzy sets, IEEE/CAA Journal of Automatica Sinica 6 (6) (2019) 1344–1351, https://doi.org/10.1109/JAS.2019.1911723.
[46] T. Lin, B.G. Horne, P. Tino, C.L. Giles, Learning long-term dependencies in NARX recurrent neural networks, IEEE Transactions on Neural Networks 7 (6) (Nov. 1996) 1329–1338.
[47] Q. Dong, Y. Lin, J. Bi, H. Yuan, An integrated deep neural network approach for large-scale water quality time series prediction, IEEE International Conference on Systems, Man, and Cybernetics (SMC 2019), Bari, Italy, October 6–9, 2019..
[48] J. Bi, Y. Lin, Q. Dong, H. Yuan, M. Zhou, An improved attention-based LSTM for multi-step dissolved oxygen prediction in water environment, in: 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), Nanjing, China, 2020.
[49] Y. Zheng, Y. Liang, S. Ke, J. Zhang, X. Yi, GeoMAN: multi-level attention networks for geo-sensory time series prediction, in: International Joint Conferences on Artificial Intelligence Organization (IJCAI), Stockholm, Sweden, 2018.
[50] S. Barra, S.M. Carta, A. Corriga, A.S. Podda, D.R. Recupero, Deep learning and time series-to-image encoding for financial forecasting, IEEE/CAA Journal of Automatica Sinica 7 (3) (2020) 683–692.