Information Sciences 625 (2023) 65–80

Contents lists available at ScienceDirect

Information Sciences

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

Multi-indicator water quality prediction with attention-assisted bidirectional LSTM and encoder-decoder ☆

Jing Bi [a], Luyao Zhang [a], Haitao Yuan [b,*], Jia Zhang [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 Computer Science in the Lyle School of Engineering at Southern Methodist University, Dallas, TX 75205, USA

ARTICLE INFO

Article history:
Received 22 August 2022
Received in revised form 9 November 2022
Accepted 29 December 2022
Available online 4 January 2023

Keywords:
Water quality prediction
LSTM
Variational modal decomposition
Particle swarm optimization
Encoder-decoder

ABSTRACT

Accurate and real-time prediction of water quality not only helps to assess the environmental quality of water, but also effectively prevents and controls water quality emergencies. In recent years, neural networks represented by Bidirectional Long Short-Term Memory (BiLSTM) and Encoder-Decoder (ED) frameworks have been shown to be suitable for prediction of time series data. However, traditional statistical methods cannot capture nonlinear characteristics of the water quality, and deep learning models often suffer from gradient disappearance and gradient explosion problems. This work proposes a hybrid water quality prediction method called SVABEG, which combines a Savitzky-Golay (SG) filter, Variational Mode Decomposition (VMD), an Attention mechanism, BiLSTM, an ED structure, and a hybrid algorithm called Genetic Simulated annealing-based Particle Swarm Optimization (GSPSO). SVABEG first adopts the SG filter and VMD to remove noise and deal with nonlinear features in the original time series, respectively. Then, SVABEG combines BiLSTM, the ED structure and the attention mechanism to capture bidirectional long-term correlations, realize dimensionality reduction and extract key information, respectively. Furthermore, SVABEG adopts GSPSO to optimize its hyperparameters. Experimental results with real-life datasets demonstrate that the proposed SVABEG outperforms current state-of-the-art algorithms in terms of prediction accuracy.

1. Introduction

With the development of cities and the acceleration of urbanization, the healthy circulation of urban water is an important basis for the healthy development of cities. It is also a basic condition for maintaining a good urban water environment and the necessary premise for maintaining a healthy urban ecological environment [1]. The quality of water quality is closely related to the safety of people's lives and properties. For a long time, the water environment has been a special concern in China. The lack of water resources in China further reflects the necessity of water resources protection. Therefore, it is very necessary to establish an accurate water quality prediction system to improve water quality. In fact, water quality prediction is a time series prediction problem [2]. In recent years, with the rapid development of Internet of Things technologies and artificial intelligence, water quality detection sensors with excellent performance, good stability and small delay emerge. By

☆ This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62073005 and 62173013, and the Fundamental Research Funds for the Central Universities under Grant YWF-22-L-1203.

* Corresponding author.

https://doi.org/10.1016/j.ins.2022.12.091
0020-0255/© 2023 Elsevier Inc. All rights reserved.

collecting the water quality information of various water quality sensors, we can obtain various water quality monitoring indicators in a real-time manner [3].

However, how to perceive change trends of water quality in advance to predict the quality of water has become a challenge [4]. Through the multi-indicator time series data collected by high-frequency water quality sensors and its correlation by mathematical models, we can extract the characteristics of time series and analyze the change trend of water quality. However, it is difficult to effectively extract characteristics of multivariate time series data for the water quality prediction [5]. At the same time, more prediction steps are needed to improve the prediction accuracy. Traditional water quality prediction methods need huge and complex parameters, and the deviation of any parameter leads to low-accuracy prediction result [6].

At present, time series forecasting methods are mainly divided into traditional statistical methods and deep learning ones. Traditional statistical methods extract linear relations of the data, and develop many excellent models. For example, Support Vector Machine (SVM) is widely used in time series forecasting due to fast speed and small amount of calculation [7]. Yet, it is difficult to cope with large-scale training samples. AutoRegressive Integrated Moving Average (ARIMA)[8] combines the advantages of linear and nonlinear models, and extracts nonlinear relations in time series. However, water quality changes are affected by many factors, and its time series is complex and nonlinear. Thus, traditional statistical methods alone cannot capture subtle changes in the water quality. Emerging deep learning techniques are widely applied to solve time series forecasting problems. Recurrent Neural Networks (RNNs) capture long-term correlations [9]. As a typical variant, long short-term memory (LSTM) solves problems of gradient disappearance and gradient explosion during long-sequence training [10]. Yet, it cannot derive causes of given outcomes. Artificial neural network models are also commonly used for time series forecasting [11], but it requires a large number of parameters and long learning time.

Although LSTM has good performance in time series prediction, it is shown that bidirectional LSTM (BiLSTM) performs better in time series prediction in some cases [12]. In addition, the Encoder-Decoder (ED) [13] provides a effective encoding and decoding mechanism to process long-term sequences and reduce their dimensionality. In addition, an attention mechanism [14] focuses on the key information in the time series and filters out useless information. Furthermore, a Savitzky-Golay (SG) filter [15] can smooth the data to remove noise in the sequence, and a variational modal decomposition (VMD) method has excellent performance in reducing and decomposing highly complex and strongly nonlinear time series into relatively stable subsequences [16]. Besides, there are many hyperparameters that affect the prediction performance, and they need to be carefully selected. Existing studies show that meta-heuristic optimization algorithms can be applied to adjust hyperparameters of training models, *e.g.*, particle swarm optimization (PSO) [17]. The prediction performance of other state-of-the-art algorithms is relatively low due to their relatively simple structures. In our algorithm, the SG filter and the VMD method are used to preprocess it to make the data have more feature dimensions. We innovatively combine BiLSTM into the ED structure to capture bidirectional dependence. Besides, Genetic Simulated annealing-based Particle Swarm Optimization (GSPSO) is used to greatly reduce the tedious process of manual parameter setting, and the attention mechanism is further used to process the data, thus forming a prediction model of water quality time series with higher precision. Therefore, to improve the prediction accuracy of the water quality time series, this work proposes a hybrid model called SVABEG, which combines the SG filter, VMD, the Attention mechanism, BiLSTM, the ED structure, and GSPSO. Main contributions of this work are summarized as:

(1) SVABEG adopts the SG filter to denoise the data to realize the data preprocessing. Then, SVABEG adopts the VMD method to handle non-stationary time series. Specifically, the time series is decomposed with VMD to yield relatively stable subsequences.
(2) SVABEG combines merits of the attention mechanism, BiLSTM, ED and GSPSO, serving as capturing key information, investigating long-term dependencies, realizing feature extraction and dimensionality reduction, and optimizing hyperparameters, respectively.
(3) Real-life data-based experimental results demonstrate that our proposed SVABEG outperforms its several typical peers in terms of prediction accuracy.

For clarity, we summarize major differences between this work and our prior one [18] as follows.

1. Different from [18], in this work, the noise signal in the time series is further smoothed locally, and the SG filter is adopted to perform weighted filtering on the original data in sliding windows, which effectively retains the change information of the original signal while maintaining the smoothness.
2. Different from [18], after the linear layer and the BiLSTM one, this work further adds the attention mechanism to extract the key features in the time series.
3. The work in [18] only adopts a type of water quality data from the Merced River of San Joaquin River, USA. Different from it, this work adopts two types of real-world datasets from multiple automatic stations in rivers of the Beijing-Tianjin-Hebei region and the Merced River of the San Joaquin River, USA, to demonstrate the prediction performance of the proposed SVABEG. For each dataset, we combine the characteristics of the time series data as the input.

The remainder of the work is organized as follows. We describe the related work in Section 2 and propose the proposed method in Section 3. The experimental results and discussions are presented in Section 4. Finally, Section 5 draws the conclusion.

2. Related Work

2.1. Classical Prediction Methods

Accurate water quality prediction plays an important role in protecting and maintaining the health of the water environment. Recent studies have shown that traditional time series forecasting methods have been widely used in various fields. They predict linear relations among data by analyzing and extracting data. Typical models such as ARIMA, support vector regression (SVR), logistic regression (LR) and back propagation (BP). The study in [19] realizes short-term passenger flow forecasting in the subway. It considers linear characteristics of the time series by combining dynamic fluctuations and ARIMA to obtain the expected passenger flow. However, it assumes that the time series data has to be stable, and relations among fluctuating time series data cannot be captured because ARIMA only captures linear relations among data.

To capture nonlinear characteristics in the time series data, several studies have shown that nonlinear models that deal with more complex data are widely used to forecast nonlinear time series. Among them, SVR, LR and BP are representative and have been widely used in various fields. The study in [20] first adopts a time series data decomposition method called Ensemble Empirical Mode Decomposition (EEMD). It then puts the decomposed data into the SVR model for training, and finally applies the gray wolf optimization algorithm (GWO) to optimize the parameters of SVR. Then, a hybrid model called GWO-SVR is proposed to predict the remaining useful life of lithium-ion batteries. In addition, the study in [21] adopts the SVR model optimized by the artificial bee colony algorithm to further predict the remaining service life of lithium-ion batteries, and the experiments demonstrate that the prediction accuracy is greatly improved. The study in [22] proposes a negative stacking framework that utilizes LR models to train weak learners in drugs, and combines them together to predict new samples, thereby improving the utilization of negative samples in drug targets. The study in [23] first obtains the individual numerical interval of the time series, and adopts a linear extraction method to obtain the change trend in the time series data. Finally, the change trend is then processed by a BP neural network for training to realize the long-term prediction of the time series. However, although above-mentioned prediction methods deal with nonlinear features in the time series data, they still do not perform well in the processing of the large-scale time series data.

2.2. Deep Learning Methods

Due to emerging technologies of big data and artificial intelligence, deep learning [24] has become an important tool to analyze data with its powerful feature extraction and data-driven abilities. Deep learning-based models are also increasingly adopted in the time series forecasting, such as Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), RNN and LSTM models. They are widely used for prediction in various areas, such as trajectory prediction, wind speed prediction, water level prediction, signal prediction, *etc.*

The study in [25] proposes a three-dimensional CNN structure characterized by spatial pyramid pooling, which not only solves a variable length problem in the time series, but also makes complex relations in patient medical records easier to capture, thereby more effectively predicting patient risks. The study in [26] integrates an improved GRU model with a resource separator module. It takes advantage of GRU's ability to effectively suppress gradient disappearance, and has low computational complexity. Then, a predictive framework is formed to further predict future resource requests. The study in [27] adopts RNN as the Encoder and Decoder in the ED structure, respectively, and adds an attention mechanism. The model adopts a sequence-to-sequence learning mechanism in the time series data to provide earthquake early warning with historical time series signals. Although RNN has excellent performance in the time series prediction, it cannot solve a problem of long-term dependence in the time series data because it only has short-term memory. As one of its variants, LSTM combines short-term and long-term memories through a special gate structure, which solves a problem of gradient disappearance to a certain extent. The study in [28] adopts LSTM as the Encoder and Decoder in the ED structure, respectively. It makes full use of the advantages of both of them, which are used to predict dynamic network links. The study in [29] proposes a hybrid model that adopts LSTM for feature extraction. Then, it adopts stacked autoencoders to encode the time series data, and adopts a multi-task learning mechanism to extract dynamic relations in the time series, thereby finally predicting the level of urban PM 2.5. The study in [30] applies LSTM for the global training, and combines multi-season decomposition techniques to propose a decomposition-based prediction framework. However, a problem with modeling with LSTM is that it cannot capture bidirectional semantic dependencies.

Different from the above studies, we innovatively combine the attention mechanism, BiLSTM and the ED structure. Specifically, we adopt BiLSTM as the encoder and decoder of the ED structure, and make full use of their respective advantages to improve the prediction accuracy of the time series.

3. Proposed Methodology

This section gives the details of our SVABEG model. First, we introduce the SG filter in 3.1 and the VMD decomposition method in 3.2. We then describe our proposed model in detail in 3.3. Finally, we present the hyperparameter tuning with GSPSO in 3.4. A loss function used in the training process is described in 3.5.

3.1. SG Filter

The SG filter is a filtering technique that adopts a fit of polynomial least squares in the time domain. It can smoothly denoise time series signals while preserving their original features. Specifically, it fits continuous subsets of adjacent data points through the process of convolution. The SG filter has two extremely important parameters, *i.e.*, the filter window size [31] and the polynomial fitting order. The window size is affected by the number of convolution coefficients in the convolution operation. It is assumed that a signal sample $z[n]$ is a sub sequence with a window size of $2m+1$, *i.e.*, $n = 2m+1$ [32]. Then, the N-order polynomial $q(n)$ used to fit data points in the window is defined as:

$$q(n) = \sum_{m=0}^{N} a_m n^m \tag{1}$$

where a_m denotes the m-th coefficient of the filter.

We minimize the following function value as much as possible, *i.e.*,

$$\hat{\delta}(n) = \sum_{n=-m}^{m} (q(n) - z[n])^2 \tag{2}$$

The output value of the polynomial is adopted as the output of the filter. Then, the data samples in the window are updated by changing a sample, and this operation is repeated to determine the next output value of the filter, which is described as:

$$y(m) = \sum_{n=-m}^{m} \bar{w}_n z_{m-n} = \sum_{n=-m}^{m} \bar{w}_{m-n} z_n \tag{3}$$

where $\bar{w}_n$ denotes the fixed impulse response of the SG filter.

Different polynomial coefficients correspond to different values of $\hat{\delta}(n)$. To determine the best coefficients of $q(n)$, we set the derivative to 0. It is represented by $n+1$ equations and $n+1$ unknown coefficients.

$$\sum_{n=0}^{N} \left(\sum_{n=-m}^{m} n^{i+m} a_m \right) = \sum_{n=-m}^{m} n^i z[n], i = 0, 1, \cdots, N \tag{4}$$

The matrix form of (4) is expressed as:

$$\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\hat{\mathbf{a}} = \mathbf{A}^{\mathrm{T}}\mathbf{z} \tag{5}$$

where $\hat{\mathbf{a}}$ denotes a polynomial coefficient vector, which consists of $\hat{\mathbf{a}} = [\hat{a}_0, \hat{a}_1, \ldots, \hat{a}_n]^{\mathrm{T}}$. $\mathbf{z}$ denotes an input sample vector, which consists of $\mathbf{z} = [z_{-m}, \ldots, z_{-1}, z_0, z_1, \ldots, z_m]^{\mathrm{T}}$. The matrix $\mathbf{A}$ is represented as:

$$\mathbf{A} = \begin{bmatrix} 1 & -m & (-m)^2 & \cdots & (-m)^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & m & (m)^2 & \cdots & (m)^n \end{bmatrix} \tag{6}$$

Therefore, the coefficient vector $\hat{\mathbf{a}}$ is expressed as:

$$\hat{\mathbf{a}} = \widetilde{\mathbf{A}}\mathbf{z} \tag{7}$$

where $\widetilde{\mathbf{A}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}$. $\widetilde{\mathbf{A}}$ is only affected by the window size and the order of the filter.

3.2. VMD

The water quality time series is relatively complex and extremely unstable, and therefore, it is difficult to predict it directly. Thus, we adopt a time series decomposition method to decompose it into several simpler subsequences. Traditional decomposition methods such as empirical mode decomposition (EMD) [33] lacks a rigorous mathematical theory that suffers from mode aliasing problems. It is shown that EEMD [34] can solve such problems. However, its calculation of multiple EMDs leads to high computational complexity. To solve above problems, this work first adopts the VMD method to decompose the original time series. It decomposes the original complex sequence into multiple relatively stable sub-sequences with a non-recursive processing strategy, thereby reducing noise interference and facilitating further prediction. As an adaptive time–frequency analysis method, VMD [35] is widely used for nonlinear and non-stationary signal decomposition. It combines the Wiener filtering algorithm, the Hilbert transform and the frequency aliasing. It can reduce fluctuations in nonlinear and noisy time series, and avoid multimodal mixing problems. Compared with other signal decomposition methods, the number of its decomposed modal components can be dynamically adjusted. These components are sparse and fluctuate widely around the center frequency, and the signal stability is achieved by minimizing the sum of the bandwidths of each modal component. The steps of calculating signal bandwidth of each modal component are given as follows.

3.2.1. Determining unidirectional spectrum

For each mode, the relevant analysis signal is calculated by the Hilbert transform to obtain the unidirectional spectrum. K denotes the number of mode components. $\xi'_k(t)$ denotes a unidirectional spectrum of the k-th $(k = 1, 2, \ldots, K)$ mode component, which is given as:

$$\xi'_k(t) = \left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t) \tag{8}$$

where $\delta(t)$ denotes a Dirichlet function and $*$ denotes a convolution operation. $\frac{j}{\pi t}$ denotes the result of the Fourier transform after convolving the original signal with another one. $u_k(t)$ denotes the k-th bandwidth-limited eigenmode function with tighter constraints.

3.2.2. Generating baseband

Each mode is mixed with the exponential frequency, and it is tuned to the corresponding estimated center frequency to move the spectrum of the mode to the baseband.

$$\xi_k(t) = \left[\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t)\right] \exp(-j\omega_k t) \tag{9}$$

where ω_k denotes an instantaneous center frequency of the k-th mode and $\xi_k(t)$ denotes a modulated modal function.

3.2.3. Calculating total bandwidth

The bandwidth of each modal component signal is estimated by the Gaussian smoothness of the demodulated signal, *i.e.*, the square norm of the gradient. The total bandwidth of all modal component signals is calculated. The variational constraint problem of VMD is given by:

$$\min_{\{u_k(t)\}\{\omega_k(t)\}} \left\{ \sum_k ||\partial_t \left[\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t)\right] \exp(-j\omega_k t)||_2^2 \right\} \tag{10}$$

$$\text{s.t.} \sum_k^K u_k(t) = x(t) \tag{11}$$

where ∂_t denotes a partial derivative operation, $x(t)$ denotes an original signal, and $||_2^2$ denotes the L^2-norm.

3.3. Proposed Model

ED is a commonly used framework for realizing time series prediction, which is composed of an encoder and a decoder. The encoder treats the input sequence as a vector with semantics, and the decoder adopts the vector as the input to decode the target sequence. For the input vector $\langle X, Y \rangle$, our goal is to predict the target Y through the ED framework, which is the predicted value given the input sequence of X. In the ED framework, other models, *e.g.*, RNN and LSTM, can also be adopted as encoders and decoders.

RNN models often suffer from gradient disappearance and gradient explosion problems due to the difficulty of training long-term sequences [36]. LSTM can solve the problems generated by RNN to a certain extent through its unique gate structure. Thus, it is often adopted as the encoder and decoder in the ED framework. An LSTM consists of three gates and a cell memory state. The hidden vectors are represented by $\left\{\mu_1, \mu_2, \cdots, \mu_\chi\right\}$. The calculation of a cell state is given as:

$$\eta = \frac{\mu_{\tau-1}}{\lambda_\tau} \tag{12}$$

$$\epsilon_\tau = \zeta(\Gamma_\epsilon \cdot \eta + \vartheta_\epsilon) \tag{13}$$

$$\beta_\tau = \zeta(\Gamma_\beta \cdot \eta + \vartheta_\beta) \tag{14}$$

$$o_\tau = \zeta(\Gamma_o \cdot \eta + \vartheta_o) \tag{15}$$

$$\kappa_\tau = \epsilon_\tau \odot \kappa_{\tau-1} + \beta_\tau \odot \tan\mu(\Gamma_\kappa \odot \eta + \vartheta_\kappa) \tag{16}$$

$$\mu_\tau = o_\tau \odot \tan\mu(\kappa_\tau) \tag{17}$$

where $\Gamma_\epsilon, \Gamma_\beta$, and Γ_o are weight matrices, and $\vartheta_\epsilon, \vartheta_\beta, \vartheta_o$ are the biases of the LSTM cell during training, which are regarded as parameters of the input, forget and output gates, respectively. ζ denotes the sigmoid function and $\odot$ denotes the element-wise multiplication. λ_τ denotes the word embedding and μ_τ denotes the hidden vector.

However, LSTM cannot capture the reverse relationship in the input information, resulting in the lack of some key information in the actual prediction. To solve this problem, we adopt BiLSTM that consists of two separate LSTMs in the forward and reverse directions. In BiLSTM, at time τ, the forward LSTM computes the hidden vector $\epsilon\mu_\tau$ based on the previous hidden vector $\epsilon\mu_{\tau-1}$ and the input word embedding λ_τ. The reverse LSTM computes the hidden vector $\psi\mu_\tau$ based on the opposite hidden vector $\psi\mu_{\tau-1}$ and the input word embedding λ_τ. Finally, $\epsilon\mu_\tau$ and $\psi\mu_\tau$ are merged into the final hidden vector, μ_τ, which is given as:

$$\mu_\tau = [\epsilon\mu_\tau, \psi\mu_\tau] \tag{18}$$

BiLSTM cannot only model contextual information in natural language processing tasks, but also effectively solve the reverse encoding problem of LSTMs. It can capture bidirectional semantic dependencies and achieve better performance on finer-grained classification tasks.

It is worth noting that when the input time series is very long, it is difficult to learn the reasonable vector representation. The attention mechanism can break the restriction that the ED structure relies on a vector with fixed length in the encoding and decoding. Thus, it is widely adopted in deep learning, and plays a prominent role in the improvement of time series learning. It enables neural networks to focus their limited attentions on the important information of the sequence. In this way, it extracts the most representative features of the time series, and pays less attention to the irrelevant information in the sequence. By adding the attention mechanism into the ED framework, the time series data can be weighted and transformed for improving the system performance.

Generally, a neural network system presents data as a set of numerical vectors with the same weight, weakening the differences among features in the data. Different from it, the attention mechanism assigns different weights to different features, and it ranks data according to their relevance [37]. Its main principle is to calculate the weight of the input element, and the element with a higher grade is assigned to a higher weight. The attention layer consists of three parts, *i.e.*, an alignment layer, the attention weight and the context vector. First, for the encoding vector $\hat{h} = \left\{\hat{h}_1, \hat{h}_2, \cdots, \hat{h}_n\right\}$ and the vertex one $\hat{\upsilon}$, we calculate their alignment score. Then, we adopt the function of softmax to normalize $\hat{h}_n$ and calculate its probability distribution $\acute{\alpha}_i$ $(i = 1, 2, \cdots, n)$. A larger value of $\acute{\alpha}_i$ indicates a larger weight, *i.e.*, the information provided by $\hat{h}_i$ is more important. Finally, the weighted sum of all elements in $\hat{h}$ is calculated, which is the output O of the attention mechanism.

$$\acute{\alpha}_i = \frac{exp\left(\hat{h}_i' \hat{\upsilon}\right)}{\sum_{j=1}^{n} exp\left(\hat{h}_j' \hat{\upsilon}\right)} \tag{19}$$

where $\hat{h}_i'$ and $\hat{h}_j'$ denote two different encoding vectors.

$$O = \sum_{i=1}^{n} \acute{\alpha}_i \hat{h}_i \tag{20}$$

To take full advantages of ED, BiLSTM and the attention mechanism, this work chooses BiLSTM as encoder and decoder, and adds the attention mechanism, thus forming the SVABEG model. The structure diagram of the SVABEG model is shown in Fig. 1. In Fig. 1, $\{f_1, f_2, \cdots, f_h\}$ denotes the water quality time series data, which are smoothed by the SG filter, and then the processed data are decomposed into relatively stable subsequences with VMD. The data dimension is then changed by a linear layer. The processed time series data are denoted by $\{f_1', f_2', \cdots, f_h'\}$, and an encoder composed of BiLSTM is used to yield the encoded sequence data $\{f_1'', f_2'', \cdots, f_h''\}$, which are then decoded by the decoder composed of BiLSTM. Then, the attention mechanism is added, and a fully connected layer is used to combine the data features obtained in the above process. Finally, prediction results of the water quality time series data are yielded.

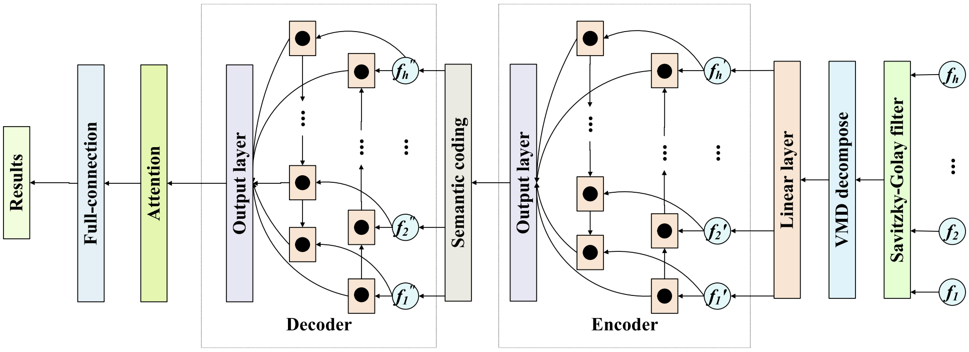

Fig. 1. Structure diagram of SVABEG. f_h' denotes the input at time step h in the Encoder, and f_h'' denotes that at time step h in the Decoder.

3.4. GSPSO

There are a large number of hyperparameters in our proposed model, and they include the number of layers, the batch size, the learning rate, the dropout rate, the weight decay, the number of epochs, and the sequence length. To make the prediction accuracy higher, a large number of studies have proven that adjusting the hyperparameters with PSO can increase the prediction accuracy. This work selects an improved version of PSO called GSPSO to optimize the hyperparameters.

PSO finds the optimal solution through cooperation and information sharing among individuals in the swarm. Its advantages are simplicity and easy implementation. At present, it has been widely used in neural network training, fuzzy system control and other application fields. In standard PSO, each particle learns from its own individual extremum and the global extremum found by the entire particle swarm to update its velocity and position. $V_i = [v_{i,1}, v_{i,2}, \cdots, v_{i,D}]$ and $L_i = [l_{i,1}, l_{i,2}, \cdots, l_{i,D}]$ denote the velocity and position of the ith particle (i = 1,2,$\cdots$, M), where M denotes the population size, respectively. $P_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,D}]$ denotes the individual extremum of particle i, and $G = [g_1, g_2, \ldots, g_D]$ denotes the global extremum of the whole particle swarm. $v_{i,d}$ and $l_{i,d}$ denote the dth entries of V_i and L_i, respectively, and they are obtained as:

$$v_{i,d} = \hat{b} \cdot v_{i,d} + c_1 \cdot r_{1,d} \cdot (p_{i,d} - l_{i,d}) + c_2 \cdot r_{2,d} \cdot (g_d - l_{i,d}) \tag{21}$$

$$l_{i,d} = l_{i,d} + v_{i,d} \tag{22}$$

$$\hat{b} = b_{max} - \frac{b_{max} - b_{min}}{\alpha_{max}} \cdot \alpha \tag{23}$$

where $\hat{b}$ denotes the inertia weight, c_1 and c_2 are acceleration coefficients that determine the relative importance of P_i and G, and $r_{1,d}$ and $r_{2,d}$ are random numbers uniformly selected in (0,1). b_{max} and b_{min} are maximum and minimum values of $\hat{b}$, respectively. α and α_{max} denote the current iteration number and its maximum limit, respectively.

Each particle learns from both P_i and G, but if both P_i and G are in the same local optimum, it may trap into the same solution all the time, which may lead to premature convergence. To solve this problem, GSPSO constructs a phenotypic combination sample $E_i = [e_{i,1}, e_{i,2}, \cdots, e_{i,D}]$ for each particle i to guide particles. $e_{i,d}$ is a linear combination of $p_{i,d}$ and g_d, which is used to change the velocity of each particle as follows.

$$v_{i,d} = \hat{b} \cdot v_{i,d} + c \cdot r_d \cdot (e_{i,d} - l_{i,d}) \tag{24}$$

$$e_{i,d} = \frac{c_1 \cdot r_{1,d} \cdot p_{i,d} + c_2 \cdot r_{2,d} \cdot g_d}{c_1 \cdot r_{1,d} + c_2 \cdot r_{2,d}} \tag{25}$$

For example, the fully informed particle swarm optimization algorithm [38] guides a particle to learn from all its neighbors, and a sample vector includes linear combinations of all locally best individuals in the neighbors. In addition, in comprehensive learning PSO [39], $e_{i,d}$ is set to $p_{i,d}$ within a predefined probability range.

3.5. Training Procedure

The loss function is used to calculate the difference between the predicted value and the ground truth one. In SVABEG, to increase the prediction accuracy, the mean square error (MSE) [40] is used as the loss function to reduce the difference between the ground truth value y_a and the predicted one $\hat{y}_a$. Smaller MSE means higher prediction accuracy of SVABEG. MSE is defined as:

$$\mathrm{MSE} = \frac{1}{Q}\sum_{a=1}^{Q}(y_a - \hat{y}_a)^2 \tag{26}$$

where Q denotes the number of training samples.

4. Experimental Evaluation

4.1. Dataset Description

We adopt two real-life datasets to evaluate the accuracy of our proposed SVABEG. The first dataset includes the data of ammoniacal nitrogen (AM), which is collected from rivers in multiple automatic platforms of the Beijing-Tianjin-Hebei region from Sept. 2018 to Dec. 2021. The second dataset includes the data of Dissolved Oxygen (DO), which is provided by the National Water Information System of the U.S. Geological Survey in the Merced River of San Joaquin River in Newman, California from May 2012 to Aug. 2020.

In the first dataset, we have a total of 7,100 pieces of samples. We choose the AM data of the Wucun platform as the ground truth value, and that of other platforms as the features. For the second dataset, there are 70,000 samples in total. It has strong periodicity, and therefore, we add the time dimension as an input feature, and divide it into three granularities, *i.e.*, month, day, and hour. Then, we have four features including month, day, hour and the amount of DO to predict the future DO in the water quality. Sensors usually collect samples at intervals of 15 to 60 min, and the data in each time interval denotes the amount of DO in this time period. For two datasets, their ratios of training, validation and test sets are set to 8:1:1.

4.2. Data Preprocessing

First, we normalize the AM and DO time series data and keep them in a unified range of (0,1) without destroying their data distribution. The formula for the normalization is shown in (27). Then, the processed data is smoothed by the SG filter, and the data is reconstructed by adjusting two important parameters, *i.e.*, the window size (w) and the order (R), thereby removing the noise in the original time series data while retaining good local characteristics. Among them, the value of w has great influence on the filtering results. Too large w makes the filtering result smooth but deviated from the ground truth value to some extent. Too small w makes the filtering result closer to the ground truth value, but it leads to relatively high noise. Similarly, the choice of R has to be reasonable. Too small R leads to fast convergence with poor steady-state performance while too large R leads to slow convergence with good steady-state performance.

$$\widetilde{Z} = \frac{Z - Z_{min}}{Z_{max} - Z_{min}} \tag{27}$$

where $\widetilde{Z}$ denotes the normalized data, Z denotes the original data, and Z_{min} and Z_{max} denote the minimum and maximum values of Z, respectively.

Thus, we first fix R to observe the loss of the time series after smoothing data with different w. Then, we fix w to observe that with different R. The optimal w and R in the SG filter are determined compared with the original time series. According to Figs. 2 and 3, when $w = 7$ and $R = 5$, the loss of the smoothed time series is the smallest, and therefore, we take such setting of w and R for the final SG filter when predicting the AM time series. Fig. 4 shows the final AM time series after denoising, which is taken as the ground truth value in its prediction. Similarly, according to Figs. 5 and 6, when $w = 9$ and $R = 5$, the loss of the smoothed time series is the smallest, and therefore, we take such setting of w and R for the final SG filter when

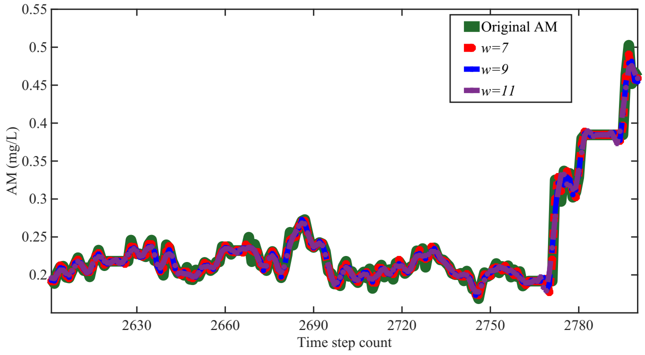

Fig. 2. The smoothed AM time series with different w.

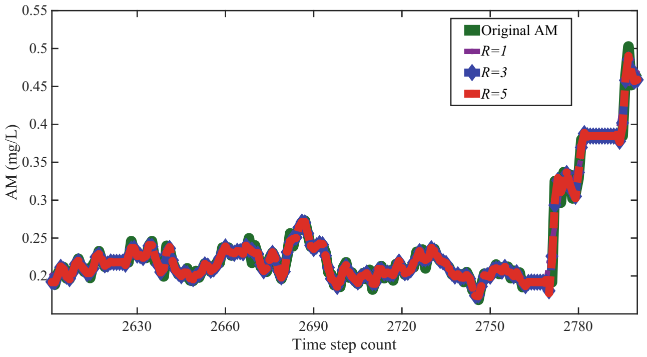

Fig. 3. The smoothed AM time series with different *R*.

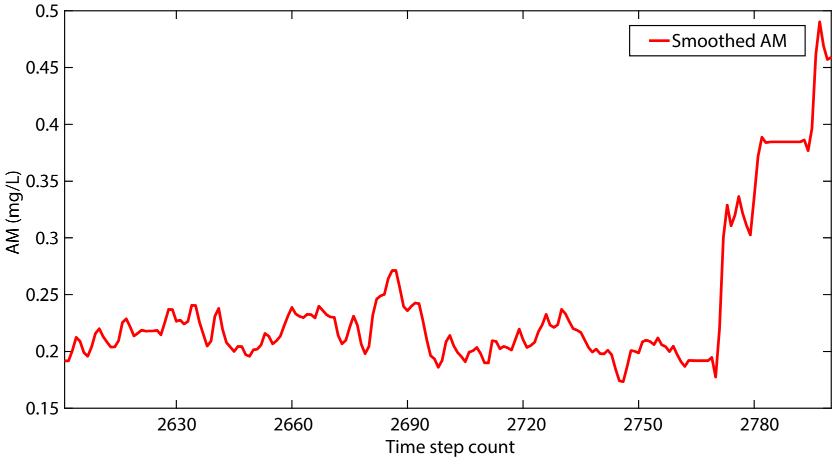

Fig. 4. The AM time series denoised by the SG filter.

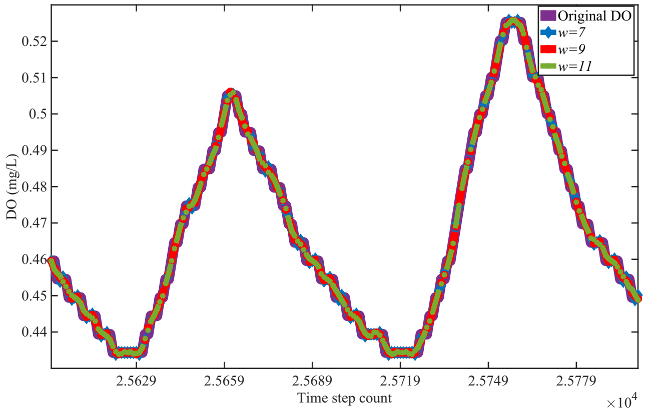

Fig. 5. The smoothed DO time series with different *w*.

predicting the DO time series. Fig. 7 shows the final DO time series after denoising, which is taken as the ground truth value in its prediction.

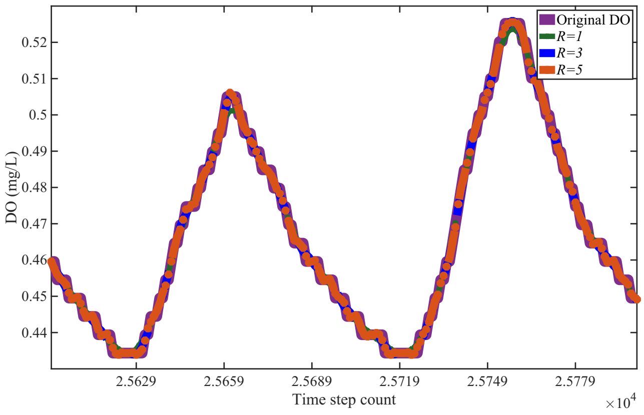

Fig. 6. The smoothed DO time series with different *R*.

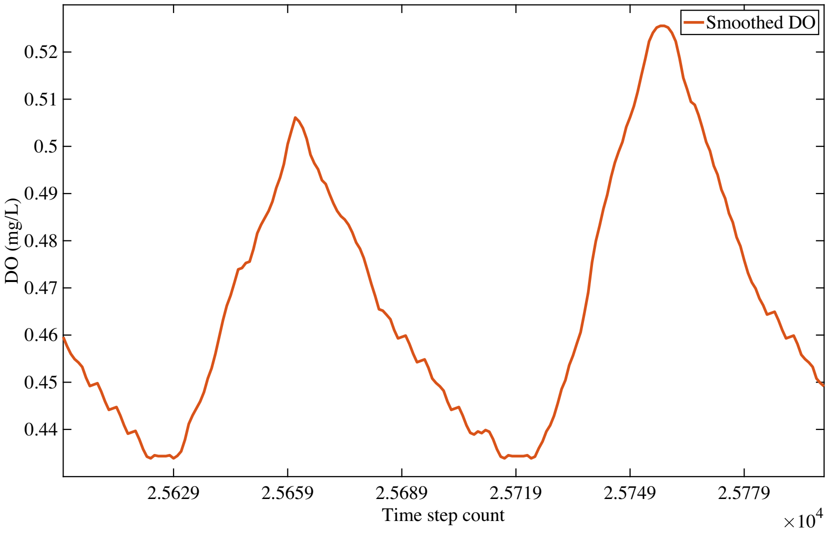

Fig. 7. The DO time series denoised by the SG filter.

4.3. Evaluation Metrics

To evaluate the prediction ability of SVABEG and other models on time series data, we adopt four metrics to evaluate the difference between the predicted value and the ground truth one. The evaluation metrics include mean absolute percentage error (MAPE) [41], mean absolute error (MAE) [42], root mean square error (RMSE) [43] and R-squared (R^2) [44]. They are given as:

$$\mathrm{MAPE} = \frac{100\%}{Q}\sum_{a=1}^{Q}\left|\frac{\hat{y}_a - y_a}{y_a}\right| \tag{28}$$

$$\mathrm{MAE} = \frac{1}{Q}\sum_{a=1}^{Q}|\hat{y}_a - y_a| \tag{29}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{Q}\sum_{a=1}^{Q}(\hat{y}_a - y_a)^2} \tag{30}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{a=1}^{Q}(y_a - \hat{y}_a)^2}{\sum_{a=1}^{Q}(y_a - \bar{y}_a)^2} \tag{31}$$

where $\bar{y}_a$ denotes the mean of the ground truth values in the sample.

4.4. Baseline Methods

This work compares the BiLSTM-ED model with several typical baseline methods including SVR, LSTM, and LSTM-ED. We compare the 5-step prediction errors of SVR, LSTM, LSTM-ED and BiLSTM-ED with respect to MAPE, MAE, RMSE and R^2. Tables 1 and 2 show experimental results of two datasets, respectively. Results show that the BiLSTM-ED model achieves higher prediction accuracy for each time series.

4.5. Parameter Tuning

Following the above BiLSTM-ED model, we adopt the SG filter to denoise the original data, decompose it with VMD, and introduce the attention mechanism. The batch size ($\imath$), the data dimension of output of the linear layer (ϱ) and the data dimension of LSTM output (ϕ) are three very important parameters in this model. Too large $\imath, \varrho$ and ϕ lead to easy convergence to local optima, while too small $\imath, \varrho$ and ϕ lead to extremely slow network training speed without convergence. Therefore, it is essential to determine appropriate values of $\imath, \varrho$ and ϕ. In the experiment, we set ϱ=ϕ. Then, we compare the loss when $\imath \in \{16, 32, 64, 128\}$ and ϱ=$\phi \in \{32, 64, 128, 256\}$ in terms of MAPE, MAE, RMSE and R^2, respectively. Tables 3 and 5 show that in the AM dataset, the loss is the smallest when $\imath$=128, and ϱ=ϕ=64. Tables 4 and 6 show that in the DO dataset, the loss is the smallest when $\imath$=64, and ϱ=ϕ=64. Thus, we choose them as the final setting of training parameters.

To improve the prediction accuracy, we adopt GSPSO to optimize other hyperparameters of our model. Here, we select three important hyperparameters including the number of layers ($\varkappa$), the dropout rate (ϖ) and the sequence length (Ψ), which are optimized by GSPSO. Finally, in the AM dataset, $\varkappa$ is 1, ϖ is 0.73, and Ψ is 30. In the DO dataset, $\varkappa$ is 2, ϖ is 0.21, and Ψ is 50. In addition, we also compare the loss with hyperparameters optimized by GSPSO with those predicted by other settings of hyperparameters. Table 7 and Table 8 shows the experimental results after selecting six different combination hyperparameters for the AM dataset and the DO dataset, respectively. It is shown that the setting of hyperparameters optimized by GSPSO has the highest prediction accuracy among all different combinations of hyperparameters.

The optimizer continuously reduces the loss by updating parameters according to their gradients. We compare four typical optimizer algorithms, *i.e.*, Stochastic Gradient Descent (SGD), Adaptive Delta (Adadelta), Adaptive Gradient (Adagrad), and Adaptive Moment Estimation (Adam). Fig. 8 shows the comparison of loss values for each optimizer as the number of training iterations increases. It is shown that Adam reduces loss values and increases the convergence speed. Therefore, Adam is finally chosen as the optimizer in SVABEG.

4.6. Comparison Of Prediction Models

According to the parameter settings, we establish the SVABEG model and fit our training samples. Fig. 9 shows the prediction results of SVABEG in the first dataset. It is shown that there is a good fit between the predicted AM and the ground truth one. In addition, we further verify the accuracy of SVABEG with the second dataset. Fig. 10 shows its predicted results of DO. The prediction results and execution speeds of other comparison algorithms are given in the supplementary file.

To further verify the effectiveness and robustness of SVABEG, we adopt MAPE, MAE, RMSE, and R^2 to compare it with SVR, BP, LSTM, BiLSTM, EMDLSTM [45], EMDBiLSTM, STLLSTM [46] and STLBiLSTM. Tables 9 and 10 show the prediction error results of above models for the AM and DO datasets, respectively. These evaluation indicators reflect the estimation of the overall deviation between the predicted value and the ground truth one. It is shown that SVABEG achieves higher pre-

Table 1
Comparison of multi-step prediction of SVR, LSTM, LSTM-ED and BiLSTM-ED for the AM dataset.

Prediction steps	SVR				LSTM				LSTM-ED				**BiLSTM-ED**			
	MAPE	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	**MAPE**	**MAE**	**RMSE**	$\mathbf{R^2}$
1	513.47	1.11	0.13	0.63	79.45	0.08	0.11	0.92	25.56	0.06	0.09	0.92	**16.23**	**0.05**	**0.07**	**0.93**
2	514.65	1.15	0.14	0.63	95.21	0.08	0.13	0.90	33.48	0.06	0.09	0.91	**16.69**	**0.05**	**0.07**	**0.91**
3	515.29	1.17	0.14	0.61	104.56	0.09	0.13	0.85	41.29	0.07	0.10	0.88	**17.20**	**0.07**	**0.08**	**0.91**
4	517.37	1.18	0.18	0.60	112.35	0.10	0.16	0.83	48.57	0.08	0.11	0.85	**17.64**	**0.08**	**0.09**	**0.90**
5	519.05	1.21	0.19	0.59	119.15	0.10	0.17	0.82	55.29	0.10	0.13	0.82	**18.54**	**0.09**	**0.09**	**0.90**
Average value	515.97	1.16	0.16	0.61	102.14	0.09	0.14	0.86	40.84	0.07	0.10	0.88	**17.26**	**0.07**	**0.08**	**0.91**

Table 2
Comparison of multi-step prediction of SVR, LSTM, LSTM-ED and BiLSTM-ED for the DO dataset.

Prediction steps	SVR				LSTM				LSTM-ED				**BiLSTM-ED**			
	MAPE	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	**MAPE**	**MAE**	**RMSE**	$\mathbf{R^2}$
1	15.13	1.30	1.50	0.70	10.71	0.91	1.15	0.82	6.92	0.56	0.79	0.83	**4.11**	**0.37**	**0.58**	**0.86**
2	14.34	1.22	1.41	0.68	10.96	0.94	1.18	0.82	7.07	0.57	0.80	0.82	**4.21**	**0.38**	**0.60**	**0.85**
3	13.61	1.16	1.33	0.68	11.34	0.97	1.22	0.81	7.21	0.59	0.81	0.82	**4.39**	**0.40**	**0.62**	**0.85**
4	13.05	1.10	1.26	0.65	11.77	1.01	1.27	0.80	7.28	0.62	0.83	0.81	**4.62**	**0.42**	**0.66**	**0.84**
5	12.53	1.05	1.19	0.65	12.20	1.05	1.32	0.77	7.35	0.64	0.83	0.80	**4.91**	**0.45**	**0.70**	**0.82**
Average value	13.73	1.17	1.34	0.67	11.40	0.98	1.23	0.80	7.17	0.60	0.81	0.82	**4.45**	**0.40**	**0.63**	**0.84**

Table 3
Prediction accuracy under different values of ι for the AM dataset.

ι	MAPE	MAE	RMSE	R^2
16	20.34	0.12	0.16	0.84
32	19.73	0.10	0.15	0.88
64	19.23	0.12	0.16	0.87
128	**18.82**	**0.05**	**0.11**	**0.90**

Table 5
Prediction accuracy under different values of ϱ and ϕ for the AM dataset.

$\varrho(\phi)$	MAPE	MAE	RMSE	R^2
32	17.94	0.06	0.11	0.85
64	**17.47**	**0.05**	**0.09**	**0.85**
128	18.32	0.14	0.21	0.76
256	18.53	0.17	0.20	0.83

Table 4
Prediction accuracy under different values of ι for the DO dataset.

ι	MAPE	MAE	RMSE	R^2
16	4.93	0.27	0.32	0.85
32	4.82	0.31	0.31	0.84
64	**4.29**	**0.23**	**0.30**	**0.86**
128	4.37	0.29	0.34	0.84

Table 6
Prediction accuracy under different values of ϱ and ϕ for the DO dataset.

$\varrho(\phi)$	MAPE	MAE	RMSE	R^2
32	4.33	0.20	0.24	0.85
64	**4.17**	**0.14**	**0.15**	**0.87**
128	4.25	0.19	0.20	0.86
256	4.29	0.17	0.21	0.86

Table 7
Comparison of different combinations of hyperparameters for the AM dataset.

Combinations	$\varkappa$	ϖ	Ψ	MAPE	MAE	RMSE	R^2
Combination 1	1	0.73	12	16.72	0.10	0.12	0.84
Combination 2	1	0.06	30	17.64	0.09	0.15	0.82
Combination 3	2	0.73	30	17.29	0.09	0.12	0.82
Combination 4	1	0.06	12	16.59	0.10	0.15	0.80
Combination 5	2	0.06	30	16.84	0.11	0.17	0.75
Combination 6	2	0.06	12	17.26	0.09	0.14	0.83
Combination by GSPSO	**1**	**0.73**	**30**	**15.78**	**0.03**	**0.04**	**0.89**

Table 8
Comparison of different combinations of hyperparameters for the DO dataset.

Combinations	$\varkappa$	ϖ	Ψ	MAPE	MAE	RMSE	R^2
Combination 1	1	0.41	32	4.35	0.09	0.08	0.87
Combination 2	1	0.41	50	4.37	0.10	0.06	0.86
Combination 3	2	0.41	32	4.34	0.09	0.08	0.86
Combination 4	2	0.41	50	4.36	0.13	0.10	0.88
Combination 5	1	0.21	50	4.37	0.10	0.07	0.89
Combination 6	2	0.21	32	4.33	0.11	0.06	0.90
Combination by GSPSO	**2**	**0.21**	**50**	**4.14**	**0.05**	**0.05**	**0.93**

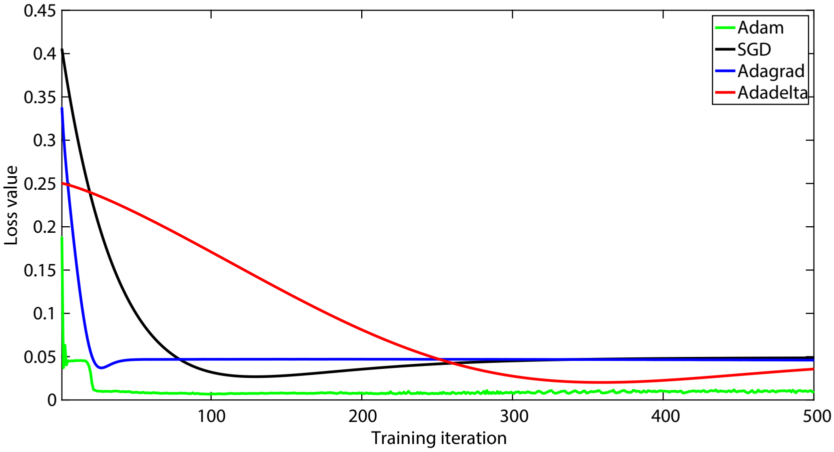

Fig. 8. Loss values of different optimizers.

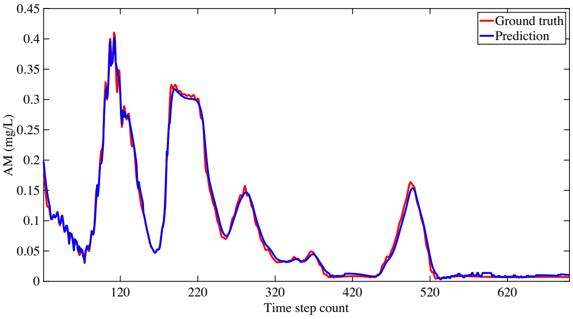

Fig. 9. AM prediction result of SVABEG.

diction accuracy than other models in terms of MAPE, MAE, RMSE and R^2. The reason is that the SG filter and VMD are used to denoise and decompose the data respectively, the attention mechanism is used to capture key information, and GSPSO is used to adjust the hyperparameters of SVABEG.

5. Conclusions and future work

Accurate prediction of water quality is of great significance to the water environment protection and health maintenance. However, water quality time series shows strong instability and nonlinear characteristics. Therefore, it cannot be accurately predicted by traditional statistical methods and a single deep learning model. To tackle this challenge, this work for the first time proposes a comprehensive prediction model named SVABEG, which combines a Savitzky-Golay (SG) filter, Variational Mode Decomposition (VMD), an Attention mechanism, Bidirectional Long Short-Term Memory (BiLSTM), an Encoder-Decoder (ED) framework, and a hybrid algorithm named Genetic Simulated annealing-based Particle Swarm Optimization (GSPSO). Specifically, we first adopt the SG filter and VMD to denoise and deal with nonlinear characteristics of the water quality time series data, respectively. Then, the attention mechanism, BiLSTM and the ED structure are adopted to retain

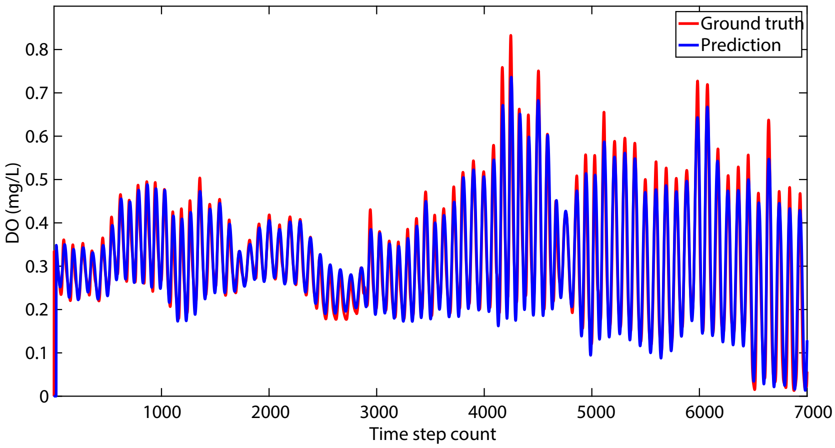

Fig. 10. DO prediction result of SVABEG.

Table 9
Comparison of different prediction models for the AM dataset.

ModelsEvaluation Metrics	MAPE	MAE	RMSE	R^2
SVR	471.23	0.18	0.22	0.87
BP	502.26	0.16	0.25	0.85
LSTM	60.42	0.09	0.14	0.89
BiLSTM	59.45	0.08	0.14	0.89
EMDLSTM	50.23	0.20	0.27	0.92
EMDBiLSTM	48.10	0.22	0.33	0.93
STLLSTM	55.20	0.07	0.14	0.94
STLBiLSTM	53.27	0.07	0.13	0.92
SVABEG	**15.64**	**0.03**	**0.04**	**0.96**

Table 10
Comparison of different prediction models for the DO dataset.

ModelsEvaluation Metrics	MAPE	MAE	RMSE	R^2
SVR	13.29	0.10	0.12	0.70
BP	15.96	0.09	0.10	0.68
LSTM	11.21	0.08	0.06	0.82
BiLSTM	12.14	0.07	0.07	0.84
EMDLSTM	11.40	0.05	0.08	0.89
EMDBiLSTM	10.35	0.06	0.08	0.89
STLLSTM	11.27	0.05	0.07	0.88
STLBiLSTM	11.23	0.05	0.06	0.90
SVABEG	**4.02**	**0.03**	**0.04**	**0.94**

the important information in the time series, capture the long-term dependencies, and reduce dimension, respectively. Finally, GSPSO is adopted to adjust hyperparameters in SVABEG to achieve higher prediction accuracy. Experimental results based on two different real-world datasets demonstrate that compared with other typical prediction models, the proposed SVABEG obtains the best prediction results in terms of the prediction accuracy.

Currently, we assume that the data acquisition equipments have no errors and provide complete and comprehensive data. Yet in practice, they suffer from failure of data sensors, which results the data missing and affects the prediction accuracy of our model. However, in the future, we will adopt processing strategies such as interpolation to complement the data. Besides, we also plan to study more advanced time series decomposition methods to deal with nonlinear features in the time series, thereby further improving the accuracy of the prediction.

CRediT authorship contribution statement

Jing Bi: Conceptualization, Supervision, Funding acquisition, Writing - original draft. **Luyao Zhang:** Formal analysis, Methodology, Validation, Data curation, Software. **Haitao Yuan:** Resources, Project administration, Visualization, Investigation. **Jia Zhang:** Investigation, Writing - review & editing.

Data availability

Data will be made available on request.

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.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.ins.2022.12.091.

References

[1] A.S. Komarov, A. Caya, M. Buehner, L. Pogson, Assimilation of SAR Ice and Open Water Retrievals in Environment and Climate Change Canada Regional Ice-Ocean Prediction System, IEEE Transactions on Geoscience and Remote Sensing 58 (6) (Jun. 2020) 4290–4303.

[2] Y. Liu, Y. Liang, K. Ouyang, S. Liu, D.S. Rosenblum, Y. Zheng, Predicting Urban Water Quality With Ubiquitous Data - A Data-Driven Approach, IEEE Transactions on Big Data 8 (2) (Apr. 2022) 564–578.

[3] M. Niroumand-Jadidi, F. Bovolo, L. Bruzzone, Novel Spectra-Derived Features for Empirical Retrieval of Water Quality Parameters: Demonstrations for OLI, MSI, and OLCI Sensors, IEEE Transactions on Geoscience and Remote Sensing 57 (12) (Dec. 2019) 10285–10300.

[4] D. Wu, H. Wang, H. Mohammed, R. Seidu, Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception, IEEE Transactions on Sustainable Computing 5 (3) (Sept. 2020) 377–388.

[5] P.C. de Lima Silva, H.J. Sadaei, R. Ballini, F.G. Guimarães, Probabilistic Forecasting with Fuzzy Time Series, IEEE Transactions on Fuzzy Systems 28 (8) (Aug. 2020) 1771–1784.

[6] J. Guo, H. Chen, J. Zhang, S. Chen, Structure Parameter Optimized Kernel Based Online Prediction With a Generalized Optimization Strategy for Nonstationary Time Series, IEEE Transactions on Signal Processing 70 (May. 2022.) 2698–2712.

[7] Y. Li, J. Wang, X. Sun, Z. Li, M. Liu, G. Gui, Smoothing-Aided Support Vector Machine Based Nonstationary Video Traffic Prediction Towards B5G Networks, IEEE Transactions on Vehicular Technology 69 (7) (Jul. 2020) 7493–7502.

[8] G. Sun, L. Song, H. Yu, V. Chang, X. Du, M. Guizani, V2V Routing in a VANET Based on the Autoregressive Integrated Moving Average Model, IEEE Transactions on Vehicular Technology 68 (1) (Jan. 2019) 908–922.

[9] M. Xia, H. Shao, X. Ma, C.W. de Silva, A Stacked GRU-RNN-Based Approach for Predicting Renewable Energy and Electricity Load for Smart Grid Operation, IEEE Trans. on Industrial Informatics 17 (10) (Oct. 2021) 7050–7059.

[10] H. Xue, D.Q. Huynh, M. Reynolds, PoPPL: Pedestrian Trajectory Prediction by LSTM with Automatic Route Class Clustering, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (Jan. 2021) 77–90.

[11] J. Bruinsma, R. Carloni, IMU-Based Deep Neural Networks: Prediction of Locomotor and Transition Intentions of an Osseointegrated Transfemoral Amputee, IEEE Transactions on Neural Systems and Rehabilitation Engineering 29 (Jun. 2021) 1079–1088.

[12] J. Sun, W. Shi, Z. Yang, J. Yang, G. Gui, Behavioral Modeling and Linearization of Wideband RF Power Amplifiers Using BiLSTM Networks for 5G Wireless Systems, IEEE Transactions on Vehicular Technology 68 (11) (Nov. 2019) 10348–10356.

[13] S. Yang, L. Li, S. Wang, W. Zhang, Q. Huang, Q. Tian, Graph Regularized Encoder-Decoder Networks for Image Representation Learning, IEEE Transactions on Multimedia 23 (Sept. 2021) 3124–3136.

[14] Q. Lai, S. Khan, Y. Nie, H. Sun, J. Shen, L. Shao, Understanding More About Human and Machine Attention in Deep Neural Networks, IEEE Transactions on Multimedia 23 (Jul. 2021) 2086–2099.

[15] A. John, J. Sadasivan, C.S. Seelamantula, Adaptive Savitzky-Golay Filtering in Non-Gaussian Noise, IEEE Transactions on Signal Processing 69 (Aug. 2021) 5021–5036.

[16] Y. Li, S. Wang, Y. Wei, Q. Zhu, A New Hybrid VMD-ICSS-BiGRU Approach for Gold Futures Price Forecasting and Algorithmic Trading, IEEE Transactions on Computational Social Systems 8 (6) (Dec. 2021) 1357–1368.

[17] P. Li, X. Liu, H. Chen, B. Li, T. Ma, W. Jiang, Optimization of Three-Dimensional Magnetic Field in Vacuum Interrupter Using Particle Swarm Optimization Algorithm, IEEE Transactions on Applied Superconductivity 31 (8) (Nov. 2021) 1–4.

[18] L. Zhang, J. Bi, H. Yuan, J. Zhang, J. Qiao, Hybrid Water Quality Prediction with Bidirectional Long Short-Term Memory and Encoder-Decoder, in: Proc. International Conference on Systems, Man, and Cybernetics, Prague, Czech Republic, 2022, pp. 1–6.

[19] C. Ding, J. Duan, Y. Zhang, X. Wu, G. Yu, Using an ARIMA-GARCH Modeling Approach to Improve Subway Short-Term Ridership Forecasting Accounting for Dynamic Volatility, IEEE Transactions on Intelligent Transportation Systems 19 (4) (Apr. 2018) 1054–1064.

[20] Z. Yang, Y. Wang, C. Kong, Remaining Useful Life Prediction of Lithium-Ion Batteries Based on a Mixture of Ensemble Empirical Mode Decomposition and GWO-SVR Model, IEEE Transactions on Instrumentation and Measurement 70 (Nov. 2021) 1–11.

[21] Y. Wang, Y. Ni, S. Lu, J. Wang, X. Zhang, Remaining Useful Life Prediction of Lithium-Ion Batteries Using Support Vector Regression Optimized by Artificial Bee Colony, IEEE Transactions on Vehicular Technology 68 (10) (Oct. 2019) 9543–9553.

[22] J. Yang, S. He, Z. Zhang, X. Bo, NegStacking: Drug-Target Interaction Prediction Based on Ensemble Learning and Logistic Regression, IEEE Transactions on Computational Biology and Bioinformatics 18 (6) (Nov. 2021) 2624–2634.

[23] W. Wang, W. Liu, H. Chen, Information Granules-Based BP Neural Network for Long-Term Prediction of Time Series, IEEE Transactions on Fuzzy Systems 29 (10) (Oct. 2021) 2975–2987.

[24] S. Roy et al, Deep Learning for Classification and Localization of COVID-19 Markers in Point-of-Care Lung Ultrasound, IEEE Transactions on Medical Imaging 39 (8) (Aug. 2020) 2676–2687.

[25] R. Ju et al, 3D-CNN-SPP: A Patient Risk Prediction System From Electronic Health Records via 3D CNN and Spatial Pyramid Pooling, IEEE Transactions on Emerging Topics in Computational Intelligence 5 (2) (Apr. 2021) 247–261.

[26] Y. Lu, L. Liu, J. Panneerselvam, B. Yuan, J. Gu, N. Antonopoulos, A GRU-Based Prediction Framework for Intelligent Resource Management at Cloud Data Centres in the Age of 5G, IEEE Transactions on Cognitive Communications and Networking 6 (2) (Jun. 2020) 486–498.

[27] T. Li, Y. Pan, K. Tong, C.E. Ventura, C.W. de Silva, Attention-Based Sequence-to-Sequence Learning for Online Structural Response Forecasting Under Seismic Excitation, IEEE Transactions on Systems, Man, and Cybernetics: Systems 52 (4) (Apr. 2022) 2184–2200.

[28] J. Chen et al, E-LSTM-D: A Deep Learning Framework for Dynamic Network Link Prediction, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (6) (Jun. 2021) 3699–3712.

[29] X. Xu, M. Yoneda, Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model, IEEE Transactions on Cybernetics 51 (5) (May 2021) 2577–2586.

[30] K. Bandara, C. Bergmeir, H. Hewamalage, LSTM-MSNet: Leveraging Forecasts on Sets of Related Time Series with Multiple Seasonal Patterns, IEEE Transactions on Neural Networks and Learning Systems 32 (4) (Apr. 2021) 1586–1599.
[31] M. Sadeghi, F. Behnia, R. Amiri, Window Selection of the Savitzky-Golay Filters for Signal Recovery From Noisy Measurements, IEEE Transactions on Instrumentation and Measurement 69 (8) (Aug. 2020) 5418–5427.
[32] A. John, J. Sadasivan, C.S. Seelamantula, Adaptive Savitzky-Golay Filtering in Non-Gaussian Noise, IEEE Transactions on Signal Processing 69 (Aug. 2021) 5021–5036.
[33] S. Gul, M.F. Siddiqui and N. u. Rehman, ‘‘FPGA-Based Design for Online Computation of Multivariate Empirical Mode Decomposition, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 12, pp. 5040–5050, Dec. 2020.
[34] X. Fan et al, Application of Ensemble Empirical Mode Decomposition in Low-Frequency Lightning Electric Field Signal Analysis and Lightning Location, IEEE Transactions on Geoscience and Remote Sensing 59 (1) (Jan. 2021) 86–100.
[35] Y. Guo, Z. Zhang, Generalized Variational Mode Decomposition: A Multiscale and Fixed-Frequency Decomposition Algorithm, IEEE Transactions on Instrumentation and Measurement 70 (Apr. 2021) 1–13.
[36] Yeqi Liu et al, DSTP-RNN: A Dual-stage Two-phase Attention-based Recurrent Neural Network for Long-term and Multivariate Time Series Prediction, Expert Systems with Applications 143 (Apr. 2020).
[37] M. Fazil, A.K. Sah, M. Abulaish, DeepSBD: A Deep Neural Network Model With Attention Mechanism for SocialBot Detection, IEEE Transactions on Information Forensics and Security 16 (Aug. 2021) 4211–4223.
[38] R. Mendes, J. Kennedy, J. Neves, The Fully Informed Particle Swarm: Simpler, Maybe Better, IEEE Transactions on Evolutionary Computation 8 (3) (Jun. 2004) 204–210.
[39] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions, IEEE Transactions on Evolutionary Computation 10 (3) (Jun. 2006) 281–295.
[40] N. Zhang, J. Ni, J. Chen, Z. Li, Steady-State Mean-Square Error Performance Analysis of the Tensor LMS Algorithm, IEEE Transactions on Circuits and Systems II: Express Briefs 68 (3) (Mar. 2021) 1043–1047.
[41] J. Vizcarrondo, J. Aguilar, E. Exposito, A. Subias, MAPE-K as a Service-oriented Architecture, IEEE Latin America Transactions 15 (6) (Jun. 2017) 1163–1175.
[42] J. Qi, J. Du, S.M. Siniscalchi, X. Ma, C.-H. Lee, Analyzing Upper Bounds on Mean Absolute Errors for Deep Neural Network-Based Vector-to-Vector Regression, IEEE Transactions on Signal Processing 68 (May. 2020.) 3411–3422.
[43] L. Somappa, A.G. Menon, A.K. Singh, A.A. Seshia, M. Shojaei Baghini, A Portable System With 0.1-ppm RMSE Resolution for 1–10 MHz Resonant MEMS Frequency Measurement, IEEE Transactions on Instrumentation and Measurement, Sept. 69 (9) (2020) 7146–7157.
[44] J. Bi, Y. Lin, Q. Dong, H. Yuan, M. Zhou, Large-scale Water Quality Prediction with Integrated Deep Neural Network, Information Sciences 571 (2021) 191–205.
[45] R. Guo, Y. Wang, H. Zhang, G. Zhang, Remaining Useful Life Prediction for Rolling Bearings Using EMD-RISI-LSTM, IEEE Transactions on Instrumentation and Measurement 70 (Jan. 2021) 1–12.
[46] X. Zhang, L. Tang, J. Chen, Fault Diagnosis for Electro-Mechanical Actuators Based on STL-HSTA-GRU and SM, IEEE Transactions on Instrumentation and Measurement 70 (Nov. 2021) 1–16.