Prediction model of sparse autoencoder-based bidirectional LSTM for wastewater flow rate

Jianying Huang1 · Seunghyeok Yang1 · Jinhui Li1 · Jeill Oh2 · Hoon Kang1

Accepted: 7 September 2022 / Published online: 26 September 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Sanitary sewer overflows caused by excessive rainfall derived infiltration and inflow is the major challenge currently faced by municipal administrations, and therefore, the ability to correctly predict the wastewater state of the sanitary sewage system in advance is especially significant. In this paper, we present the design of the Sparse Autoencoder-based Bidirectional long short-term memory (SAE-BLSTM) network model, a model built on Sparse Autoencoder (SAE) and Bidirectional long short-term memory (BLSTM) networks to predict the wastewater flow rate in a sanitary sewer system. This network model consists of a data preprocessing segment, the SAE network segment, and the BLSTM network segment. The SAE is capable of performing data dimensionality reduction on high-dimensional original input feature data from which it can extract sparse potential features from the aforementioned high-dimensional original input feature data. The potential features extracted by the SAE hidden layer are concatenated with the smooth historical wastewater flow rate features to create an augmented previous feature vector that more accurately predicts the wastewater flow rate. These augmented previous features are applied to the BLSTM network to predict the future wastewater flow rate. Thus, this network model combines two kinds of abilities, SAE’s low-dimensional nonlinear representation for original input feature data and BLSTM’s time series prediction for wastewater flow rate. Then, we conducted extensive experiments on the SAE-BLSTM network model utilizing the real-world hydrological time series datasets and employing advanced SVM, FCN, GRU, LSTM, and BLSTM models as comparison algorithms. The experimental results show that our proposed SAE-BLSTM model consistently outperforms the advanced comparison models. Specifically, we selected a 3 months period training dataset in our dataset to train and test the SAE-BLSTM network model. The SAE-BLSTM network model yielded the lowest RMSE, MAE, and highest $R^2$, which are 242.55, 179.05, and 0.99626, respectively.
Keywords  Time series prediction · Sparse autoencoder · Bidirectional long short-term memory · Data dimensionality reduction · Sanitary sewer overflows (SSOs) · Rainfall derived infiltration and inflow (RDII)

1 Introduction

Sanitary sewer overflows (SSOs) are one of the major environmental problems faced by the current municipal administration institutions [1]. In addition to waste derived from seasonal production and everyday needs of people, the large-scale gathering of people (such as the Olympic Games, World Expo, etc.) and prolonged epidemics (for example, Coronavirus-19) exist. All of these events flush a large amount of wastewater into sanitary sewer systems, resulting in wastewater levels exceeding processing capacity and sanitary sewer overflows. Rainfall derived inflow and infiltration (RDII) is another significant cause of excessive wastewater, or sanitary sewer overflows in the sanitary sewer system. RDII relates to external rainfall and groundwater entering the sanitary sewer system, either as rain flowing into drains or as groundwater infiltrating by various means into the sanitary sewage system, leading to an excess of wastewater in the sanitary sewage system [2]. Excessive RDII can lead to serious operational problems, both for sewage pipes and the sewage treatment plants themselves. More specifically, it may also increase the cost of sewage treatment, increase the burden on the pipes carrying the sewage, increase the risk of an SSO occurring, and increase the risk of sewer pipe collapse [3]. There are also risks to human health and the ecological environment when too much RDII causes wastewater from sanitary sewer systems to overflow into streets, buildings, and even flood public and private property. If we want to reduce the risk and danger of SSOs caused by excessive RDII into the sanitary sewer system, it is significant to accurately predict the wastewater state of the sanitary sewer system in advance of RDII. Furthermore, the ability to accurately predict the wastewater state of the sanitary sewer system would be an invaluable tool in the spheres of urban management, sewage treatment plant design, planning, and optimization [4].

Given that flow information can accurately assess the inflow and infiltration of sewage in the sanitary sewage system, the use of wastewater flow rates, organized in time series, will allow for RDII to be better estimated. Predicting the wastewater flow rate in the sanitary sewage system is a hydrological time series prediction problem. In the past, researchers have resorted to applying a variety of traditional mathematical models to predict the hydrological time series. These traditional mathematical models can be divided into linear models and nonlinear models according to their types. With regard to the linear models, ARMA (Auto Regressive Moving Average) [5], ARIMA (Auto-Regressive Integrated Moving Average) [6], and SARIMA (Seasonal Autoregressive Integrated Moving Average) were widely utilized to predict the hydrological time series [7–9]. These linear models assume that the data is fixed, the ability to capture nonlinearity is limited, and that a nonlinear process cannot be simulated. In an attempt to address the complex nonlinear hydrological time series prediction problem, some researchers
have applied some nonlinear models. These researchers [10–15] employed SVM (Support Vector Machine) or ANN (Artificial Neural Network) in machine learning algorithms [16] to more accurately predict hydrological time series. Although the prediction results of nonlinear models have been found to be better than linear models, the accuracy and performance of the prediction when confronted with a highly complex non-stationary hydrological time series remains deficient in certain aspects.

In recent years, more and more researchers have begun applying deep learning algorithms to predict hydrological time series. Compared with the aforementioned traditional algorithms, deep learning algorithms have stronger nonlinear simulation capabilities. Thus far, deep learning algorithms have excelled in the areas of CV (Computer Vision) [17, 18] and NLP (Natural Language Processing) [19, 20], where they are playing an increasingly important role. The strength of deep learning algorithms is that they excel at simultaneously processing both complex nonlinear data and big data. The deep learning algorithms can analyze the correlation between multivariable, discover the potential features of their internal structure, compress the amount of the original input feature data and by doing this reduce computational complexity, and so help improve the accuracy of prediction. The Autoencoder (AE) [21] and Sparse Autoencoder (SAE) networks [22] are such deep learning algorithms. Because of this, some researchers [23, 24] adopted AE networks to compress high-dimensional hydrological time series to obtain low-dimensional representations and generate predictions based on low-dimensional representations. Besides, others researchers [25–27] exploited the SAE network to extract features from hydrological time series and generate predictions based on the extracted features. Although both AE and SAE can reduce the dimensionality of high-dimensional features, the features extracted by AE are relatively redundant, and those extracted by SAE are relatively concise. Meanwhile, the deep learning algorithms can handle large-scale datasets and can maintain long-term dependencies between the time series, like the Gate Recurrent Unit (GRU) [28], Long Short-Term Memory (LSTM) [29], and Bidirectional Long Short-Term Memory (BLSTM) [30] networks. For this reason, these researchers [2, 31–35] leveraged the GRU, LSTM, and BLSTM networks to predict hydrological time series. However, GRU can ignore some important information due to its simple structure, LSTM can only consider historical information, and BLSTM can comprehensively consider historical information and future information. In addition to RNNs (Recurrent Neural Networks), CNN (Convolutional Neural Network) [36] was also applied to predict hydrological time series. But, CNN is good at analyzing spatial information and not good at analyzing temporal information. As a result, we combined the SAE network and the BLSTM network to predict our hydrological time series.

In this paper, our ultimate objective is to create a network model that can efficiently extract features, accurately predict wastewater flow rate, and significantly reduce training time. Here, we designed the Sparse Autoencoder-based Bidirectional Long Short-Term Memory (SAE-BLSTM) network model, which is built using both the SAE and the BLSTM networks. The SAE-BLSTM network model can extract sparse potential features from the original previous features of wastewater flow rate and the previous features of related variables, and then concatenate both
the extracted sparse potential features and the smooth previous wastewater flow rate features to create an augmented previous feature vector that accurately predicts the future wastewater flow rate. The SAE-BLSTM network model is divided into two parts: the SAE network and the BLSTM network. The basic principles of our network model are as follows: (1) Due to the characteristics of the SAE network, sparse potential features can be extracted from any high-dimensional multivariate time series. The features extracted by the SAE network are more concise than the original features. We may also reduce the original high-dimensional data’s dimensionality and represent the original high-dimensional input data with the obtained low-dimensional features. Hence, the SAE network not only simplifies computational complexity but also aims to increase prediction accuracy. (2) The LSTM network is a special variation of RNN (Recurrent Neural Network), which can remember previous information, and utilize the previous information, calculate the current output. In this paper, the BLSTM network is applied in the prediction phase, where the created augmented previous feature vector is input into the BLSTM network to predict future wastewater flow rates. The BLSTM network is a network connecting two LSTMs running in opposite directions between the input layer and the output layer, which means that it considers both the past and the future information within a specific range of time series data, something which helps improve prediction accuracy [37]. In general, the SAE network part effectively extracts sparse potential features from the original input data by reducing the dimensionality of the original input data, while the BLSTM network part comprehensively considers the past information and the future information that can improve the accuracy of prediction.

The following sets out the organization of this paper: Sect. 3 introduces the details of the SAE-BLSTM network model. Section 3 deals with the pre-experiment preparation, including the dataset details, the evaluation methods, and the structure of the comparison algorithms. Section 4 displays the results of the experiments. Finally, Sect. 5 outlines the results of the experiments.

2 Methods

2.1 Data preprocessing

We have deployed four types of sensors to obtain four types of data. The four types of data are the wastewater flow level ($H$), the wastewater flow speed ($V$), the wastewater flow rate ($Q$), and the rainfall ($R$), respectively. For simplicity of expression, we use $H$, $V$, $Q$, and $R$ to represent the four data types. Due to noise in the data collected by the sensors, the data waveform is not smooth enough. Thus, we applied a modified Bartlett–Hann window FIR filter as a simple moving average filter to eliminate noise, maintain the data waveform as smooth as possible, and obtain high-quality $H$, $V$, $Q$, and $R$ data. Since each input feature has a different scale, we adopted the Z-Score Normalization approach to normalizing the data, which is dependent on the mean and standard deviation of the data.

After data filtering and normalization, the smooth $H$, $V$, $Q$, and $R$ features need to be divided into the previous feature values and the corresponding labels. In this
procedure, we exploited the sliding window. The length of the sliding window represents the length of a single input feature vector at each time step, which also means that the network model will predict the next time value based on the previous features of window length [38]. In this paper, the length parameter of the sliding window is 12, which means that we used the smooth $H$, $V$, $Q$, and $R$ features in the past 12 time steps as the previous feature values. Because there are four variables in each time step, after concatenating the 12-time steps of previous feature values to create the previous feature vectors, each previous feature vector size is $(48, 1)$. For the corresponding labels, the single feature $Q$ in the $T$ time step is predicted. Besides, the single feature $Q$ in the $T+11$ time steps (11-steps ahead) is predicted.

3 Sparse autoencoder (SAE) network

The AE is a classical unsupervised learning neural network algorithm. The AE network has the functions of data dimensionality reduction and feature extraction [21]. Figure 1 shows the fundamental structure of the AE network. The forward propagation of the AE network can be split into two procedures, namely the Encoding procedure and the Decoding procedure. In the Encoding procedure, the input layer and the hidden layer are used to convert the original input data into the hidden representation code. In the Decoding procedure, the hidden layer and the output layer are used to reconstruct the original input data from the trained hidden representation code.

This AE network displays some disadvantages, such as just copying the original input data from the input layer and storing the original input data in the hidden layer. Although the output layer can perfectly reconstruct the original input data, the extracted features are redundant and meaningless. Using an SAE network can remedy the disadvantages of the AE network. The SAE network is a variant of the AE network. The SAE network adopts the sparse coding idea [39] and append the sparse penalty term based on the AE network. Under the sparse constraint, the hidden layer of the SAE network can learn the sparse and concise feature representation of the original input data from the input layer [40–42]. Adding the sparse constraint
limits the activation value of the hidden layer, thereby limiting the number of neurons in the hidden layer. Under the sparse constraint, some hidden layer neurons are active, while other neurons are inactive. The SAE network can outperform the traditional AE network and has more practical application significance.

Here, sample \( x \) comes from dataset \( X = [x_1, x_2, \ldots, x_n] \), which is composed of \( n \) samples. It is worth noting that \( x_n \) refers to the previous feature vector (48,1) that we concatenated above. The activation value \( h \) in the hidden layer of the SAE network is calculated as follows:

\[
h = f\left( W^{(1)}x + b^{(1)} \right)
\]

where \( W^{(1)} \) is the weight that connects the input and the hidden layer, \( b^{(1)} \) is the bias. \( f \) represents the activation function, and the sigmoid function \( f(x) = 1/(1+e^{-x}) \) is selected in this procedure. Then using the weight that connects the hidden and the output layer to reconstruct the hidden representation code into the original input data of the input layer. The reconstruction data is calculated as follows:

\[
\hat{x} = f\left( W^{(2)}h + b^{(2)} \right)
\]

where \( \hat{x} \) indicates the reconstructed data, \( W^{(2)} \) is the weight that connects the hidden and the output layer, and \( b^{(2)} \) is the bias. \( f \) represents the activation function, and the linear transfer function \( f(x) = x \) is selected in this procedure.

At the beginning of training the SAE network, the weights \( W^{(1)} \) and \( W^{(2)} \), and the biases \( b^{(1)} \) and \( b^{(2)} \) are initialized randomly. After the initialization of the parameters, the SAE network performs the forward propagation, calculates activation values in the hidden layer, and then uses the calculated activation values to reconstruct the original input data in the output layer. For all data \( x_i, i = 1, 2, 3, \ldots, n \) in the dataset, it is necessary to design the overall loss function of the SAE network and calculate the reconstruction error. The overall loss function of the SAE network is described as:

\[
J(W, b) = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{1}{2} ||x_i - \hat{x}_i||^2 \right) + \frac{\lambda}{2} \sum_{i=1}^{n} \sum_{j=1}^{s_{l+1}} \left( W_{ij}^{(l)} \right)^2
\]

where the loss function \( J(W, b) \) contains two parameters \( W \) and \( b \), which need to be optimized. \( n \) denotes the number of layers in the neural network, \( l \) means the serial number of the neural network layer. \( s_j \) refers to the number of neurons in the \( l \)th layer of the neural network. \( W_{ij}^{(l)} \) is all of the weight vectors of the connection between the \( l \)th layer and the \( l+1 \)th layer. In Eq. 3, the first part calculates the reconstruction error of the entire dataset and minimizes the reconstruction error. The second part is the regularization weight penalty term, which is used to prevent overfitting. \( \lambda \) indicates the weight decay parameter.

In the hidden layer of the SAE network, the average activation degree of the \( j \)th neuron over the dataset is calculated as:
\[
\rho_j = \frac{1}{n} \sum_{i=1}^{n} h_j(x_i), j = 1, \ldots, s_l
\] (4)

where \(s_l\) denotes the number of neurons in the \(l\)th hidden layer in the SAE network.

By appending a sparse penalty term \(\rho\) to limit the activation degree of each neuron in the hidden layer of the SAE network. The calculation of all node constraints in the \(l\)th hidden layer of the SAE network is:

\[
\sum_{j=1}^{s_l} KL(\rho \rho_j) = \sum_{j=1}^{s_l} \rho \log \frac{\rho}{\rho_j} + (1 - \rho) \log \frac{(1 - \rho)}{(1 - \rho_j)}
\] (5)

where \(KL\) means the Kullback–Leibler (KL) divergence, under this constraint, the activation degree \(\rho_j\) of all neurons in the hidden layer is close to the given sparsity penalty \(\rho\). After adding the sparse constraint, the loss function is redefined as:

\[
J_{\text{sparse}}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_l} KL(\rho \rho_j)
\] (6)

where \(\beta\) refers to the coefficient for the sparsity penalty term, which controls the influence of the sparsity penalty term in the cost function. The loss function needs to be minimized, and the parameters are updated by the Scaled Conjugate Gradient Algorithm [43].

Before the BLSTM network prediction, we applied the SAE network to extract potential features from the original input data and perform data dimensionality reduction on the original input data. The parameters, in particular, the number of neurons and activation functions used in the SAE network, directly affect whether effective features can be extracted, thus affecting the whole experimental result. The appropriate output feature dimensionality of the hidden layer in the SAE network and the appropriate activation function in the SAE network cannot be directly estimated. The number of neurons in the hidden layer and the activation function in the SAE network were determined experimentally to address this problem [44]. After data preprocessing, the shape of the SAE network input feature vector is (48, 1), so the number of neurons of the input layer in the SAE network is 48. To speed up the calculations and reduce reconstruction errors, we only used a hidden layer in the SAE network. When setting up the neurons of the hidden layer in the SAE network, we set up 8, 16, 24, 32, 40, and 48 neurons, respectively. For the activation function in the SAE network, we propose two strategies which are then compared. The first strategy is to use the sigmoid activation function in both the Encoding procedure and the Decoding procedure of the SAE network. The second strategy is to use the sigmoid activation function in the SAE network Encoding procedure, and the linear transfer function in the SAE network Decoding procedure.

For the experiment, we set up a validation set in our dataset. The SAE network is trained for 2000 epochs with the validation set, and the reconstruction error between the original input data and the reconstructed data is calculated using the MSE (Mean Squared Error) [45]. Figure 2 displays the SAE network’s training outcomes in different neurons and different activation functions. For the
activation function in the SAE network, we find that the reconstruction error of the second strategy is significantly lower than the first strategy, so the second strategy is selected. And for the number of neurons of the hidden layer in the SAE network, we find that when 8, 16, or 24 neurons are used, the reconstruction error is in a rapid decline stage. When the number of neurons used exceeds 24, that is, when 32, 40, and 48 neurons are used, the reconstruction error will not change much. In other words, when more than 24 neurons are used, the impact on training results is not much improved. Considered thus comprehensively, we selected 24 neurons.

Fig. 2 The reconstruction error when using different numbers of neurons and different activation functions in the hidden layer.

Fig. 3 Our proposed SAE network’s structure.
Figure 3 presents our proposed SAE network’s structure. The original input data with the shape of \((48, 1)\) is reduced to the representation features the shape of \((24, 1)\) by the Encoder of the SAE network.

### 4 BLSTM network

Compared with the RNN network, the LSTM network can address the problem of gradient vanishing while retaining the influence of the previous feature in the dataset for a longer period of time [46, 47]. However, as the LSTM network can only take the previous information in the dataset into account, it is beneficial to the prediction task if the future information in the dataset can be considered. Therefore, the BLSTM network plays an important role, since it contains two bidirectional LSTM networks, considering not only the previous information in the dataset but also the future information in the dataset.

Figure 4 demonstrates the specific structure of the BLSTM and the LSTM cell. The LSTM network overcomes the problem of gradient vanishing due to its memory cells and the logic gates. As shown in Fig. 4, the LSTM cell contains a memory cell and three logic gates. The three logic gates are the forget gate, the input gate, and the output gate.

The LSTM network is implemented by the following equations:
\[ f_t = \sigma \left( W_{hf}^{h_{t-1}} h_{t-1} + W_f^x x_t + b_f \right) \]  

\[ i_t = \sigma \left( W_{hi}^{h_{t-1}} h_{t-1} + W_i^x x_t + b_i \right) \]  

\[ c_t = f_t \odot c_{t-1} + i_t \odot \tanh \left( W_c^{h_{t-1}} h_{t-1} + W_c^x x_t + b_c \right) \]  

\[ o_t = \sigma \left( W_o^{h_{t-1}} h_{t-1} + W_o^x x_t + b_o \right) \]  

\[ h_t = o_t \odot \tanh(c_t) \]

where \( f_t, i_t, c_t, o_t, \) and \( h_t \) represent the output of the forget gate, input gate, cell state, output gate, and hidden state at the current time \( t \), respectively. \( \sigma \) and \( \tanh \) indicate the element-wise sigmoid and the element-wise hyperbolic tangent activation function. \( \odot \) denotes the element-wise product. \( W_{hf}, W_{hi}, W_c, W_o \) are the weight matrices of the forget gate, input gate, cell state, and output gate for the previous moment hidden state vector \( h_{t-1} \), and the current moment input vector \( x_t \), respectively. It is worth emphasizing that the current moment input vector \( x_t \) of the BLSTM network refers to the augmented feature vector \((36, 1)\) that we create in the following. The forget gate determines what information in the cell state is discarded or retained by Eq. 7. The input gate determines what new information is selected from the input, adds the new information to the cell state, and updates the cell state by Eqs. 8, 9. Finally, the output gate determines the final output information, including our predicted wastewater flow rate and the new hidden state by Eqs. 10, 11.

## 5 SAE-BLSTM network model

Figure 5 illustrates our designed SAE-BLSTM network model. The SAE-BLSTM network model consists of two parts. The first part is data preprocessing. First, we need to filter the multivariate features acquired by the different sensors to obtain smooth multivariate features. Our dataset has four variable features at each time step: the \( H, V, Q, \) and \( R \). After the filtering operation, we can obtain the smooth \( H, V, Q, \) and \( R \), respectively. Then, we divide the smooth multivariate features into the previous feature values and the future prediction values. We use the past 120 min (the past 12 time steps) of smooth \( H, V, Q, \) and \( R \) as previous feature values to predict the future 10 min and 110 min (11-steps ahead) \( Q \). To create the smooth previous feature vectors \((48, 1)\), we finally concatenate the four previous feature values of the past 12 time steps.

Given the smooth previous feature vectors and the future prediction values, the second part trains the hybrid network model built on the SAE and BLSTM networks to predict the wastewater flow rate. First, we input the smooth previous feature...
vectors (48, 1) into the SAE network, and then we only use the hidden layer output of the SAE network as the extracted potential feature vectors (24, 1). The potential feature vector extracted (24, 1) by the SAE network is concatenated with the single previous feature vector (12, 1) of the smooth wastewater flow rate to create an augmented previous feature vector (36, 1). After that, we divide these augmented previous feature vectors (36, 1) and the future prediction values into training, validation, and test sets.

Fig. 5 The wastewater flow rate prediction network model (SAE-BLSTM network model)
For the following BLSTM network, the outputs of the forward LSTM network and the backward LSTM network of the BLSTM network are all connected to a Fully Connected (FC) Layer with only one neuron. This FC Layer is used to compute the Root-Mean-Square Error (RMSE) of the regression output and generate predictions. Once the datasets with augmented previous feature vectors (36, 1) are obtained, we input them into the BLSTM network to predict the future wastewater flow rates. The training set is adopted to train the parameters of the BLSTM network, and the validation set is exploited to adjust the parameters of the BLSTM network and evaluate the generalization ability of the BLSTM network.

The combination of the SAE network and the BLSTM network can not only speed up the training time but also increase the accuracy of prediction.

6 Experiment

6.1 Dataset

The dataset was collected by four different sensors within the NAMBANG-4 sewage sensor point of the Yangju City Wastewater Treatment Plant in Korea. The four different sensors detected the wastewater flow level ($H$), wastewater flow speed ($V$), wastewater flow rate ($Q$), and rainfall ($R$) every 10 min. The $H$, $V$, and $Q$ reflect the level, velocity, and rate of water flow in the wastewater plant sewage pipes, respectively. The $R$ reflects the presence or absence of rain during data collection and the magnitude of rain. Table 1 displays specific details about the input variable names and their corresponding units. The time range of the dataset is 2017.08.08–2018.08.10, with a total of 364 days, and 52,416 input feature data.

7 Comparison of model structure

To measure the performance improvement of our proposed network model, we compare the SAE-BLSTM network model with the advanced machine learning model SVM and the advanced deep learning models FCN [48], GRU, LSTM, and BLSTM. We applied MATLAB R2021a to implement all models, and all deep learning models are trained based on GPU. Table 2 presents the key parameter

| Table 1  | The details of the input variables |
|----------|----------------------------------|
| Input variable names | Units |
| Wastewater flow level ($H$) | mm |
| Wastewater flow speed ($V$) | m/s |
| Wastewater flow rate ($Q$) | m$^3$/day |
| Rainfall ($R$) | mm |
settings of SVM, FCN, GRU, LSTM, BLSTM, and SAE-BLSTM models. Table 3 lists the configuration of our experimental platform.

### 8 Evaluation method

We employ the RMSE (Root-Mean-Square Error), MAE (Mean Absolute Error), and $R^2$ (Coefficient of Determination) as evaluation methods in the experiment. The three evaluation methods are implemented as follows:

\[
RMSE = \left( \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \right)^{\frac{1}{2}} \quad (12)
\]

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (13)
\]
here the number of samples in the test set is \( n \), and the true and predicted values of the wastewater flow rate are \( y_i \) and \( \hat{y}_i \), respectively. \( \bar{y} \) represents the mean of all true values in the test set.

**9 Experimental results**

First, we selected about 3 months, 91 days, of data from the dataset, the time range is 2017.08.10 AM 1:00:00–2017.11.08 AM 1:00:00, the number of input feature data is 144 (input feature data/day) \( \times \) 91 (days) = 13104 (input feature data), according to the specified ratio of [8:1:1], which is split into a training set, a validation set, and a test set. Due to the boundary conditions, the number of input previous feature vectors of the training set, validation set, and test set are 10444, 1305, and 1305, respectively.

\[
R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{14}
\]

Table 4 The relevant information about the computational complexity includes input dimension and training time for each model in the training and validation process

| Model name   | Input dimension | Training time (S) | Final training loss | Final validation RMSE |
|--------------|-----------------|-------------------|---------------------|-----------------------|
| SVM          | (48, 1)         | 4                 | –                   | –                     |
| FCN          | (48, 1)         | 128               | 0.03410             | 0.26076               |
| GRU          | (48, 1)         | 143               | 0.00927             | 0.13613               |
| LSTM         | (48, 1)         | 154               | 0.00871             | 0.13195               |
| BLSTM        | (48, 1)         | 181               | 0.01976             | 0.19880               |
| SAE-BLSTM    | (36, 1)         | SAE/BLSTM         | 73/158              | 0.00667               | 0.11547               |

Fig. 6  The comparison of the training loss and validation RMSE for each model in the training process

Table 4  The relevant information about the computational complexity includes input dimension and training time for each model in the training and validation process

Here the number of samples in the test set is \( n \), and the true and predicted values of the wastewater flow rate are \( y_i \) and \( \hat{y}_i \), respectively. \( \bar{y} \) represents the mean of all true values in the test set.
Figure 6 displays the comparison of the training loss and validation RMSE for each model in the training process. As shown in Fig. 6, we see that the application of the SAE network and the creation of augmented features contribute to stable and fast convergence.

To evaluate the computational complexity of the SAE-BLSTM network model, Table 4 summarizes the relevant information, including the input dimension and training time for each model in the training and validation process. In Table 4, the original input dimension (48, 1) is converted to (36, 1) due to the application of the SAE network and the creation of augmented features. Therefore, the SAE-BLSTM network model has the advantage of input dimension compared to other models. The SAE-BLSTM network model consists of the SAE network and the BLSTM network, and thus the training time also consists of the SAE network training time and the BLSTM network training time. For a fair comparison, the training time of the BLSTM network part of the SAE-BLSTM network model is only compared with the training time of the original BLSTM network model. As seen in Table 4, compared with the original BLSTM network model, the SAE-BLSTM network model has a smaller input dimension and less training time. Hence, the efficient and compact features can reduce computational complexity, thereby reducing prediction errors and improving prediction accuracy.

Figure 7 demonstrates the comparison of the predicted and true values, the prediction errors, and the correlation coefficients $R^2$ in the test process. As shown in Fig. 7, compared with other models, the RMSE of the SAE-BLSTM network model is decreased by approximately 58%, 46%, 27%, 19%, and 18%, respectively, and the MAE is decreased by approximately 29%, 49%, 19%, 17%, and 10%, respectively. The closer the correlation coefficient $R^2$ is to 1, the better the model prediction performance. We find that the $R^2$ exceeds 0.99 and approaches 1 for all deep learning models except the SVM model. Therefore, the difference in $R^2$ between the deep learning models is insignificant, including the $R^2$ difference between the...
Table 5  The comparison of the training time, RMSE, MAE, and $R^2$ of the test set for 6 models in the different training periods

| Number of days | SVM | FCN | GRU | LSTM | BLSTM | SAEB/LSTM | SVM | FCN | GRU | LSTM | BLSTM | SAEB/LSTM | SVM | FCN | GRU | LSTM | BLSTM | SAEB/LSTM |
|----------------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|
| 90             | 4   | 119 | 131 | 140  | 184   | 72/157    | 572.41 | 312.13 | 314.08 | 294.15 | 283.34 | **159.54** | 251.39 | 211.85 | 220.34 | 198.66 | 196.24 | **180.31** |
| 120            | 5   | 145 | 166 | 173  | 234   | 92/203    | 179.05 | 224.35 | 242.48 | 238.82 | 295.37 | **151.43** | 136.65 | 186.20 | 200.73 | 192.03 | 236.33 | **114.57** |
| 150            | 6   | 234 | 275 | 279  | 371   | 118/290   | 383.92 | 274.33 | 273.66 | 214.39 | 238.29 | **194.65** | 222.60 | 205.45 | 206.54 | 140.38 | 166.30 | **117.49** |
| 180            | 7   | 284 | 331 | 340  | 448   | 142/400   | 483.34 | 214.27 | 207.09 | 204.40 | 165.47 | **142.86** | 189.95 | 173.79 | 144.99 | 165.23 | 120.68 | **104.43** |
| 210            | 8   | 360 | 403 | 422  | 556   | 168/355   | 147.40 | 405.94 | 368.27 | 393.81 | 393.08 | **281.65** | 471.39 | 244.15 | 164.02 | 179.43 | 242.00 | **136.52** |
| 240            | 9   | 391 | 455 | 462  | 617   | 186/570   | 923.84 | 333.22 | 194.22 | 231.16 | 178.18 | **180.37** | 427.37 | 265.42 | 149.82 | 164.49 | 135.69 | **120.63** |
| 270            | 10  | 453 | 520 | 528  | 708   | 215/601   | 2470.15 | 598.97 | 495.52 | 505.22 | 452.25 | **265.79** | 740.83 | 230.45 | 241.71 | 213.04 | 190.50 | **182.18** |
| 300            | 11  | 461 | 533 | 546  | 724   | 234/678   | 4662.68 | 1444.87 | 956.24 | 1150.99 | 1033.96 | **503.67** | 1467.08 | 813.98 | 414.03 | 464.30 | 411.51 | **277.20** |
| 330            | 14  | 496 | 548 | 581  | 768   | 263/720   | 5370.32 | 886.13 | 803.75 | 767.63 | 735.68 | **454.63** | 2110.15 | 503.61 | 379.95 | 384.25 | 376.50 | **272.01** |
| 364            | 19  | 514 | 563 | 592  | 848   | 284/788   | 1999.43 | 855.20 | 826.05 | 527.08 | 512.91 | **437.57** | 797.14 | 609.06 | 552.57 | 287.82 | 292.63 | **272.69** |

The boldface letters represent our best results.
SAE-BLSTM network model and the BLSTM network model. Nevertheless, the SAE-BLSTM network model obtains the highest $R^2$. The experimental results mean that the predicted values of the SAE-BLSTM network model are closest to the true values.

Table 5 demonstrates the training time, RMSE, MAE, and $R^2$ of the test set for 6 models in different training periods of 3–12 months. No matter how many days of data are selected, the RMSE and MAE of the SAE-BLSTM network model are consistently the lowest, and the correlation coefficient $R^2$ is consistently the highest, which indicates that the prediction results of the SAE-BLSTM network model are always the best. Compared with other models, the SAE-LSTM network model not only has the best prediction results but also is the most robust. In this table, we mainly compare the performance difference between the SAE-BLSTM network model and the BLSTM network model. In prediction results, the prediction error between the two network models is relatively significant, and the correlation coefficient $R^2$ is relatively insignificant. In addition, we measured the training time to evaluate the computational complexity of our network model, and we still mainly compare the training time of the BLSTM network part of the SAE-BLSTM network model and the original BLSTM network model. In the training time, the training time of the BLSTM network part in the SAE-BLSTM network model is consistently less than the training time of the original BLSTM network, so our proposed model can always reduce the computational overhead and speed up the training process.

We divided the entire dataset into different 3 months training periods to evaluate the performance of the SAE-BLSTM network model. Table 6 illustrates the training time, RMSE, MAE, and $R^2$ of the test set for 6 models in different 3 months training periods. Regardless of the training period, the SAE-BLSTM network model always has the lowest RMSE and MAE, and the highest $R^2$, which suggests that the prediction accuracy of the SAE-BLSTM network model is always the highest. Compared with other models, the SAE-BLSTM network model consistently exhibits excellent predictive performance. In prediction results, the prediction errors between SAE-BLSTM and BLSTM network models are relatively significant, and the correlation coefficient $R^2$ are relatively insignificant. In training time, our proposed network model always plays a role in reducing computational errors and improving prediction accuracy.

Finally, we analyzed the relationship between the NAMBANG-4 sewage sensor point and the adjacent sewage sensor points. We connected the NAMBANG-4 sewage sensor point data ($H_1, V_1, Q_1, R_1$) with the MAJEON-1 sewage sensor point data ($H_2, V_2, Q_2, R_2$) or the YEYANG-2 sewage sensor point data ($H_3, V_3, Q_3, R_3$) to predict $Q_1$ (The wastewater flow rate of the NAMBANG-4 sewage sensor point). The NAMBANG-4 sewage sensor point is the downstream location, and the MAJEON-1 sewage sensor point and the YEYANG-2 sewage sensor point are the upstream locations, respectively. Figure 8 identifies the locations of the NAMBANG-4 sewage sensor point, the MAJEON-1 sewage sensor point, and the YEYANG-2 sewage sensor point on the map.

When connecting the downstream location A data with the upstream location B data or the upstream location C data, the data dimension increases sharply. The above experiments have completely verified the superior performance of the
Table 6  The comparison of the training time, RMSE, MAE, and $R^2$ of the test set for 6 models in different 3 months training periods

| Time periods           | SVM | FCN | GRU | LSTM | BLSTM | SAE-BLSTM | SVM | FCN | GRU | LSTM | BLSTM | SAE-BLSTM | SVM | FCN | GRU | LSTM | BLSTM | SAE-BLSTM | SVM | FCN | GRU | LSTM | BLSTM | SAE-BLSTM | SVM | FCN | GRU | LSTM | BLSTM | SAE-BLSTM | SVM | FCN | GRU | LSTM | BLSTM | SAE-BLSTM |
|------------------------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|-----|-----|-----|------|-------|-----------|
| 2017.08.10–2017.11.10  | 5   | 119 | 127 | 133  | 185   | 76/172    | 566.19| 352.50| 346.74| 290.31| 325.70| 226.94     | 251.23| 269.58| 201.37| 199.33| 242.21| 159.78     | 0.98010| 0.9557| 0.99280| 0.99500| 0.99440| 0.99670     |
| 2017.11.11–2018.02.11  | 5   | 111 | 128 | 136  | 183   | 76/173    | 199.93| 129.38| 117.90| 122.60| 120.33| 116.49     | 156.73| 99.17 | 89.05 | 92.30 | 91.72 | 87.81      | 0.97000| 0.99471| 0.99888| 0.99979| 0.99880| 0.99993     |
| 2018.02.12–2018.05.12  | 5   | 127 | 148 | 153  | 205   | 72/184    | 1255.84| 247.03| 251.27| 215.24| 226.92| 177.41     | 633.61| 191.35| 162.37| 151.90| 156.59| 137.71     | 0.89411| 0.99776| 0.99624| 0.99717| 0.99725| 0.99984     |
| 2018.05.13–2018.08.08  | 5   | 124 | 145 | 151  | 199   | 70/168    | 403.27| 277.63| 259.23| 209.69| 213.35| 192.67     | 282.89| 223.66| 210.20| 159.65| 161.01| 151.14     | 0.99081| 0.99724| 0.99600| 0.99747| 0.99732| 0.99763     |

The boldface letters represent our best results.
SAE-BLSTM network model. Here, we verified the difference between applying the SAE network and not applying the SAE network before the BLSTM network model, as well as verified whether the prediction of the wastewater flow rate at downstream location A is influenced by the data at upstream locations B or C.

According to the specified ratio of [8:1:1], we still divided the 364 days data into a training set, a validation set, and a test set. We used the previous feature data of the past 120 min to predict the wastewater flow rate ($Q_1$) of 110 min (11-step ahead) the future. Figure 9 illustrates the predicted and true values, RMSE, MAE, and $R^2$ of the test set for 6 experimental methods. In Fig. 9, BLSTM(A), BLSTM(A + B), and BLSTM(A + C) represent the data of location A, location A + B, and location A + C as the input of the BLSTM network, the dimensionality...
of the input data the BLSTM network are (48, 1), (96, 1), (96, 1), respectively. Similarly, the data of location A, location A + B, and location A + C as the input of the SAE-BLSTM network model, the dimensionality of the input data of the SAE-BLSTM network model are (48, 1), (96, 1), (96, 1), respectively. Before the BLSTM network, the SAE network was applied to reduce the dimensionality of the input data. To compare various algorithms fairly, we applied the SAE network to compress the data of A (48, 1), A + B (96, 1), and A + C (96, 1) into (24, 1). Finally, due to the creation of the augmented vector, the dimensions of the input data of the BLSTM network in the SAE-BLSTM network model are (36, 1), (36, 1), (36, 1), respectively.

In Fig. 9, compared with the BLSTM network model, when the SAE-BLSTM network model uses the dataset of sensor point A, the combined dataset of sensor points A and B, and the combined dataset of sensor points A and C to predict the wastewater flow rate at sensor point A, the RMSE decreases by approximately 10%, 61%, and 63%, respectively, the MAE decreases by approximately 10%, 46%, and 43%, respectively, and the $R^2$ increases by approximately 1%, 17%, and 34%. Therefore, the predicted values of the SAE-BLSTM network model are always closest to the true values.

Moreover, we observe that both the BLSTM network model and the SAE-BLSTM network model obtain better prediction results based on the combined dataset of sensor points A and B than those based on the combined dataset of sensor points A and C. The result demonstrates that the data of sensor point A combined with the dataset of the nearby sensor points can help improve the prediction accuracy. In particular, compared with those based on the dataset on only sensor point A, the SAE-BLSTM network model based on the combined dataset of sensor points A and B can help to further improve the prediction accuracy.

We also observe that the SAE-BLSTM network model based on a single dataset or a combined dataset consistently outperforms the BLSTM network model based on a single dataset or a combined dataset. For the explanation of this phenomenon, on the one hand, due to the combination of datasets, the input data dimensionality of the BLSTM network model increases sharply, both the computational complexity and computation time also increase. As a result, the cumulative prediction error increases in the back-propagation calculation, making the prediction results deteriorate. On the other hand, despite the input data's dimensionality increases sharply due to the combination of datasets. However, when we apply the SAE network before the BLSTM network model, the SAE networks can extract potential features from the original input data, reduce the input data dimensionality of the BLSTM network models, and reduce the computational complexity and computation time, while making the prediction results better. This proves that applying the SAE network before the BLSTM network has obvious advantages compared to not applying the SAE network before the BLSTM network.
10 Conclusion

This paper proposed a novel SAE-BLSTM network model to predict wastewater flow rate. Before the SAE-BLSTM network model prediction, we applied a simple moving average filter to filter the original input data, reduce noise, and obtain smooth, high-quality input data. We then input the smooth input data into the SAE network, which can extract potential features from the original input data and reduce the dimensionality of the original input data. The potential features extracted by the SAE network were concatenated with the smooth wastewater flow rate features to create an augmented previous feature vector. Finally, the augmented previous feature vector was fed into the BLSTM network to predict the wastewater flow rate. Subsequently, we performed several experiments based on the SAE-BLSTM network model and selected advanced SVM, FCN, GRU, LSTM, and BLSTM models as comparison models to demonstrate the excellent performance of the SAE-BLSTM network model. The experiment in the 3 months training period dataset demonstrates that the SAE-BLSTM network model provides a stable and fast optimization process that benefits from the augmented feature vector creation. The experiment also indicates that the SAE-BLSTM network model offers the advantages of low computational complexity, fast training speed, low prediction error, and high prediction precision that benefits from the concise and compact augmented feature vector. In the evaluation metrics, the SAE-BLSTM network model obtains the minimum RMSE, MAE, and maximum $R^2$, which are 242.55, 179.05, and 0.99626, respectively. The experiment in the different training period and 3 months training period datasets demonstrate that the SAE-BLSTM network model exhibits good generalization ability and stability in the different scale and interval datasets. The experiment in the connected upstream and downstream sensor datasets demonstrates that applying the SAE network to compress the high-dimensional datasets and extract features before the BLSTM network delivers obvious advantages in predicting results. The experiments also indicate that connecting the target sensor point dataset to an adjacent sensor point dataset, especially the nearby sensor point dataset, can help reduce prediction errors and improve prediction accuracy.

In this work, we introduce an SAE network to process hydrological time series and then generate predictions based on a type of RNN. However, the inconvenience of applying the SAE network is that we demand multiple experiments to determine the best hyper parameters in the SAE network. In general, it presents a reference for dealing with high-dimensional time series. In future, we will incorporate a spatiotemporal attention mechanism into an LSTM network model to further improve prediction accuracy.

Acknowledgements This research was supported by the Chung-Ang University Young Scientist Scholarship in 2021, by Korea Environment Industry & Technology Institute (KEITI) through Public Technology Program based on Environmental Policy (Project No: 2018000700005) funded by Korea Ministry of Environment (MOE), and by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1009735).

Data availability Not applicable.
Code availability  Not applicable.

Declarations

Conflict of interest  The authors declare that they have no conflict of interest.

References

1. Zhang Z (2007) Estimating rain derived inflow and infiltration for rainfalls of varying characteristics. J Hydraul Eng 133(1):98–105
2. Kang H, Yang S, Huang J, Oh J (2020) Time series prediction of wastewater flow rate by bidirectional LSTM deep learning. Int J Control Autom Syst 18(12):3023–3030
3. Zhang M, Liu Y, Cheng X, Zhu DZ, Shi H, Yuan Z (2018) Quantifying rainfall-derived inflow and infiltration in sanitary sewer systems based on conductivity monitoring. J Hydrol 558:174–183
4. Zeng Y, Zhang Z, Kusiak A, Tang F, Wei X (2016) Optimizing wastewater pumping system with data-driven models and a greedy electromagnetism-like algorithm. Stoch Env Res Risk Assess 30(4):1263–1275
5. Box GE, Pierce DA (1970) Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. J Am Stat Assoc 65(332):1509–1526
6. Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. Wiley, Hoboken
7. Valipour M, Banhabib ME, Behbahani SMR (2013) Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. J Hydrol 476:433–441
8. Valipour M (2015) Long-term runoff study using SARIMA and ARIMA models in the United States. Meteorol Appl 22(3):592–598
9. Martínez-Acosta L, Medrano-Barboza JP, López-Ramos Á, Remolina López JF, López-Lambrario ÁÁ (2020) SARIMA approach to generating synthetic monthly rainfall in the Sinú river watershed in Colombia. Atmosphere 11(6):602
10. Lin GF, Chen GR, Huang PY, Chou YC (2009) Support vector machine-based models for hourly reservoir inflow forecasting during typhoon-warning periods. J Hydrol 372(1–4):17–29
11. Guo J, Zhou J, Qin H, Zou Q, Li Q (2011) Monthly streamflow forecasting based on improved support vector machine model. Expert Syst Appl 38(10):13073–13081
12. Niu WJ, Feng ZK (2021) Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management. Sustain Cities Soc 64:102562
13. Tokar AS, Johnson PA (1999) Rainfall-runoff modeling using artificial neural networks. J Hydrol Eng 4(3):232–239
14. Dawson CW, Wilby RL (2001) Hydrological modelling using artificial neural networks. Prog Phys Geogr 25(1):80–108
15. Gholami V, Sahour H (2022) Simulation of rainfall-runoff process using an artificial neural network (ANN) and field plots data. Theoret Appl Climatol 147(1):87–98
16. Prasanth A (2021) Certain investigations on energy-efficient fault detection and recovery management in underwater wireless sensor networks. J Circuits Syst Comput 30(08):2150137
17. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778
18. Wang CY, Bochkovskiy A, Liao HYM (2022) YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint https://arxiv.org/abs/2207.02696
19. Collobert R, Weston J (2008, July) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. pp. 160–167
20. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2021) Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. arXiv preprint https://arxiv.org/abs/2107.13586
21. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
22. Ng A (2011) Sparse autoencoder. In: CS294A. Lecture notes, 72:1–19
23. Kao IF, Liou JY, Lee MH, Chang FJ (2021) Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts. J Hydrol 598:126371
24. Abbasi M, Farokhnia A, Bahreinimotlagh M, Roozbahani R (2021) A hybrid of Random Forest and Deep Auto-Encoder with support vector regression methods for accuracy improvement and uncertainty reduction of long-term streamflow prediction. J Hydrol 597:125717
25. Yuan Y, Jia K (2015) A water quality assessment method based on sparse autoencoder. In: 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC) (pp. 1–4). IEEE
26. Li Z, Peng F, Niu B, Li G, Wu J, Miao Z (2018) Water quality prediction model combining sparse autoencoder and LSTM network. IFAC-PapersOnLine 51(17):831–836
27. Qian L, Li J, Liu C, Tao J, Chen F (2020) River flow sequence feature extraction and prediction using an enhanced sparse autoencoder. J Hydroinf 22(5):1391–1409
28. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint https://arxiv.org/abs/1412.3555
29. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
30. Siami-Namini S, Tavakoli N, Namin AS (2019) The performance of LSTM and BiLSTM in forecasting time series. In: 2019 IEEE International Conference on Big Data (Big Data) (pp. 3285–3292). IEEE
31. Gao S, Huang Y, Zhang S, Han J, Wang G, Zhang M, Lin Q (2020) Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. J Hydrol 589:125188
32. Kratzert F, Klotz D, Brenner C, Schulz K, Herrnegger M (2018) Rainfall-runoff modelling using long short-term memory (LSTM) networks. Hydrol Earth Syst Sci 22(11):6005–6022
33. Hu C, Wu Q, Li H, Jian S, Li N, Lou Z (2018) Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water 10(11):1543
34. Sahoo BB, Jha R, Singh A, Kumar D (2019) Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting. Acta Geophys 67(5):1471–1481
35. Xiang Z, Yan J, Demir I (2020) A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resour Res. https://doi.org/10.1029/2019WR025326
36. Shu X, Ding W, Peng Y, Wang Z, Wu J, Li M (2021) Monthly streamflow forecasting using convolutional neural network. Water Resour Manag 35(15):5089–5104
37. Zhang B, Zhang H, Zhao G, Lian J (2020) Constructing a PM2.5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks. Environ Modell Softw 124:104600
38. Shen Z, Zhang Y, Lu J, Xu J, Xiao G (2020) A novel time series forecasting model with deep learning. Neurocomputing 396:302–313
39. Ols Hansen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583):607–609
40. Sun C, Ma M, Zhao Z, Tian S, Yan R, Chen X (2018) Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. IEEE Trans Industr Inf 15(4):2416–2425
41. Liu J, Li Q, Han Y, Zhang G, Meng X, Yu J, Chen W (2019) PEMFC residual life prediction using sparse autoencoder-based deep neural network. IEEE Trans Transp Electrific 5(4):1279–1293
42. Zhang K, Zhang J, Ma X, Yao C, Zhang L, Yang Y, Zhao H (2021) History matching of naturally fractured reservoirs using a deep sparse autoencoder. SPE J 26(04):1700–1721
43. Möller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6(4):525–533
44. Liu H, Chen C (2019) Multi-objective data-ensemble wind speed forecasting model with stacked sparse autoencoder and adaptive decomposition-based error correction. Appl Energy 254:113686
45. Sekar J, Aruchamy P, Sulaima Lebbe Abdul H, Mohammed AS, Khamuruddeen S (2022) An efficient clinical support system for heart disease prediction using TANFIS classifier. Comput Intell 38(2):610–640
46. Cheng Q, Chen Y, Xiao Y, Yin H, Liu W (2022) A dual-stage attention-based Bi-LSTM network for multivariate time series prediction. J Supercomput. https://doi.org/10.1007/s11227-022-04506-3
47. Gul MJ, Urfal GM, Paul A, Moon J, Rho S, Hwang E (2021) Mid-term electricity load prediction using CNN and Bi-LSTM. J Supercomput 77(10):10942–10958
48. MATLAB Deep Learning Toolbox Documentation (2021a), Fully Connected Layer. https://www.mathworks.com/help/releases/R2021a/deeplearning/ref/nnet.cnn.layer.fullyconnectedlayer.html?%3ds_tid=doc_ta

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Jianying Huang¹ · Seunghyeok Yang¹ · Jinhui Li¹ · Jeill Oh² · Hoon Kang¹

✉ Hoon Kang
hkang@cau.ac.kr
Jianying Huang
jianying@cau.ac.kr

¹ School of Electrical and Electronics Engineering, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul 06974, Korea

² Department of Civil and Environmental Engineering, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul 06974, Korea