
Research article

Enhancing sewage flow prediction using an integrated improved SSA-CNN-Transformer-BiLSTM model

Jiawen Ye[&], Lei Dai[&] and Haiying Wang^{*}

School of Science, China University of Geosciences (Beijing), Beijing 100083, China

*** Correspondence:** Email: whycht@126.com.

& These authors contributed equally to this work and should be considered co-first authors.

Abstract: Accurate prediction of sewage flow is crucial for optimizing sewage treatment processes, cutting down energy consumption, and reducing pollution incidents. Current prediction models, including traditional statistical models and machine learning models, have limited performance when handling nonlinear and high-noise data. Although deep learning models excel in time series prediction, they still face challenges such as computational complexity, overfitting, and poor performance in practical applications. Accordingly, this study proposed a combined prediction model based on an improved sparrow search algorithm (SSA), convolutional neural network (CNN), transformer, and bidirectional long short-term memory network (BiLSTM) for sewage flow prediction. Specifically, the CNN part was responsible for extracting local features from the time series, the Transformer part captured global dependencies using the attention mechanism, and the BiLSTM part performed deep temporal processing of the features. The improved SSA algorithm optimized the model's hyperparameters to improve prediction accuracy and generalization capability. The proposed model was validated on a sewage flow dataset from an actual sewage treatment plant. Experimental results showed that the introduced Transformer mechanism significantly enhanced the ability to handle long time series data, and an improved SSA algorithm effectively optimized the hyperparameter selection, improving the model's prediction accuracy and training efficiency. After introducing an improved SSA, CNN, and Transformer modules, the prediction model's R^2 increased by 0.18744, $RMSE$ (root mean square error) decreased by 114.93, and MAE (mean absolute error) decreased by 86.67. The difference between the predicted peak/trough flow and monitored peak/trough flow was within 3.6% and the predicted peak/trough flow appearance time was within 2.5 minutes away from the monitored peak/trough flow time. By employing a multi-model fusion approach, this study achieved efficient and accurate sewage flow prediction, highlighting the potential and application prospects of the model in the field of sewage treatment.

Keywords: sewage flow prediction; bidirectional long short-term memory network (BiLSTM); sparrow search algorithm (SSA); transformer; convolutional neural network (CNN)

Mathematics Subject Classification: 65K10, 68T07

1. Introduction

With the ongoing urbanization process and the constant need for water environment improvement, the demand for urban sewage treatment is increasing, placing greater operational pressure on sewage treatment plants. In this context, accurate prediction of sewage flow plays a crucial role in the operation of sewage treatment plants [1]. Timely and accurate sewage flow prediction not only helps managers better allocate resources, optimize treatment processes, and reduce energy consumption, but also aids in preventing and responding to sudden sewage treatment demands and potential environmental pollution incidents [2]. For example, during heavy rainfall, sewage treatment plants can adjust treatment processes in advance based on predicted flow changes to minimize the loss which overflow and pollution accidents may bring [3]. Moreover, changes in sewage flow not only affect the normal operation of sewage treatment facilities, but also directly relate to environmental protection and public health [4]. Therefore, establishing effective sewage flow prediction models has become an urgent issue for modern sewage treatment plants.

Currently, mainstream sewage flow prediction models mainly include traditional statistical models [5] and machine learning models [6]. Traditional statistical models such as autoregressive integrated moving average (ARIMA) and seasonal ARIMA (SARIMA) rely on the historical trends and periodic changes of time series data. They have solid theoretical foundations and simple model structures but perform limitedly when dealing with nonlinear and high-noise data [7]. Li et al. [8] adopted an ARIMA model to reduce the delay of rainfall data used as model input. This model was used to predict sewage flow in two real sewage pumping stations (SPS) with different hydraulic characteristics and climatic conditions. Liu et al. [9] employed the ARIMA method to establish an emergency prediction model for water pollution in different sections of the Qingzhang River and formulated adaptive urban river water purification strategies based on its variation patterns. Machine learning models such as support vector machines (SVM) and random forests (RF) predict by training on large amounts of historical data, demonstrating strong nonlinear fitting capabilities. Ekinci et al. [10] developed a prediction model based on machine learning algorithms to accurately and quickly predict sewage sludge. They tested the predictive performance of different machine learning algorithms using data obtained from a real advanced biological wastewater treatment plant in Kocaeli, Turkey. Machine learning models typically require large amounts of historical data for training, necessitating high-quality and large datasets, and still have limitations in handling complex spatiotemporal relationships.

Additionally, deep learning models such as long short-term memory (LSTM) and bidirectional LSTM(BiLSTM) have shown outstanding performance in time series prediction [11]. Yaqub et al. [12] proposed and developed a neural network based on LSTM to predict the removal efficiencies of ammonia (NH-N), total nitrogen (TN), and total phosphorus (TP) in an anaerobic-anoxic-oxic membrane bioreactor (A-A-O MBR) system. Farhi et al. [13] proposed a novel machine learning method based on LSTM architecture to predict anomalies in sewage treatment plants several hours in advance. Alfwzan et al. [14] enhanced water quality prediction by combining BiLSTM networks with computational fluid dynamics (CFD). This combined model leveraged the synergy between deep learning and fluid dynamics simulation to overcome the limitations of existing methods, thereby improving prediction accuracy and efficiency. Despite the excellent performance of BiLSTM in

handling time series prediction, it still faces several challenges in practical applications [15]. First, due to the need for both forward and backward computation paths, BiLSTM models have high computational and memory requirements, leading to relatively low training and inference efficiency [16]. Second, while BiLSTM models can capture long-term temporal dependencies, their complex model structure can easily lead to overfitting, especially with limited data [17]. Moreover, BiLSTM models require the complete time series data for bidirectional computation, posing latency issues for prediction needs [18]. Hence, finding a model that can enhance prediction efficiency and accuracy is particularly important.

The performance of prediction models largely depends on the selection of hyperparameters, making hyperparameter optimization a key step in improving prediction accuracy. Traditional hyperparameter optimization methods such as grid search and random search, while simple and easy to implement, are inefficient when dealing with high-dimensional and complex models [19]. Intelligent optimization algorithms like genetic algorithm (GA), particle swarm optimization (PSO), and Bayesian optimization utilize mechanisms such as evolution and swarm intelligence to find better hyperparameter combinations in a shorter time [20]. Wang et al. [21] established a Tent_BP(back propagation)_SSA-based hybrid model combining the tent chaotic map and sparrow search algorithm (SSA) for predicting the effluent quality of sewage treatment processes. Farzin et al. [22] used artificial neural networks and support vector regression (SVR) models to predict biogas production in anaerobic digesters at a municipal sewage treatment plant in southern Tehran, combining GA and PSO for model hyperparameter optimization to enhance training performance and obtain optimal input parameters. Ye et al. [23] developed models to predict effluent TN (total nitrogen) and TEC (total energy consumption) by selecting influent quality and process control indicators as input features, exploring the predictive performance of machine learning methods under different random seeds, and using moving average for data amplification and Bayesian algorithms for hyperparameter optimization. Intelligent optimization algorithms are widely used in hyperparameter optimization of machine learning models due to their strong global search capability and fast convergence [24]. However, these algorithms also face issues such as high computational complexity, susceptibility to local optima, and sensitivity to initial parameters [25]. Therefore, there is a need to develop more efficient optimization algorithms for hyperparameter optimization to further improve the training efficiency and prediction performance of sewage flow prediction models.

As mentioned above, to address research gaps, we propose an innovative prediction model integrating an enhanced SSA algorithm, CNN, Transformer, and BiLSTM for sewage flow forecasting. This novel approach utilizes CNN for extracting local features and detecting short-term patterns [51], Transformer for capturing global dependencies through its attention mechanism, and BiLSTM for advanced temporal processing of the features. The improved SSA algorithm optimizes hyperparameter selection within the CNN-Transformer-BiLSTM framework, aiming to surpass the limitations of single models. The novelty of this model is combining CNN, Transformer, and the BiLSTM model. Moreover, the introduction of the improved SSA is also an innovation. This multi-model fusion aims to enhance the prediction accuracy and generalization, offering a cutting-edge solution for efficient sewage flow prediction. The contributions of this paper are as follows:

- (1) Development of a CNN-Transformer-BiLSTM model for sewage flow prediction to address the challenges of BiLSTM in capturing global important features and overfitting.
- (2) A multi-strategy integrated improved SSA algorithm is proposed to enhance training efficiency and prediction performance.
- (3) Achieving efficient and accurate sewage flow prediction using a multi-model fusion method based on an improved SSA-CNN-Transformer-BiLSTM framework, overcoming the limitations of single models.

2. Materials and methods

2.1. Moving average smoothing

Moving average smoothing (MAS) [26] is a commonly used data preprocessing technique widely applied in signal processing and time series analysis. Its primary objective is to smooth data and reduce the influence of random noise, making the data trends more apparent. The basic idea of MAS is to calculate the average of data points within a certain range around each data point in the data sequence and replace the original data point with this average. This process slides a fixed-size window over the data sequence. We used MAS to process the sewage flow: one spanning from 07:05:00 on December 4, 2023, to 14:45:00 on January 12, 2024, and the another one spanning from 00:00:00 on April 1, 2024, to 16:35:00 on May 7, 2024. After applying MAS, the sewage flow trend became more obvious, which is conducive to more accurate prediction of the flow in the future.

Considering a data sequence $\{x_1, x_2, \dots, x_n\}$ with a sliding window size ω (typically an odd number), the smoothed value y_i for each data point x_i can be represented as:

$$y_i = \frac{1}{\omega} \sum_{j=i-\lceil \omega/2 \rceil}^{i+\lceil \omega/2 \rceil} x_j, \quad (1)$$

where $\lceil \cdot \rceil$ signifies floor division to ensure alignment of the window center point. The data sequence processed by MAS exhibits reduced fluctuations compared to the original data sequence, better reflecting the overall data trend. Overall, MAS is a fundamental and practical data smoothing method that effectively enhances data quality and interpretability in various application scenarios. Therefore, MAS algorithm is selected in this study to preprocess the wastewater flow dataset, aiming to improve the quality of predictive data.

2.2. CNN-Transformer-BiLSTM prediction model

The combined prediction model constructed in this study integrates CNN [27], BiLSTM [28], and Transformer [29] architecture, referred to as the CNN-Transformer-BiLSTM model. The architecture of this model is illustrated in Figure 1, comprising convolutional layers, pooling layers, Transformer layers, BiLSTM layers, and fully connected layers. Combining the advantages of these three methods, we can obtain more powerful sequence modeling capabilities, from local to global and from one-way to two-way, and improve the performance of the model in an all-round way.

The CNN is utilized to capture local features and detect short-term patterns in sewage flow data, processing spatial hierarchies to identify critical features influencing flow patterns. However, CNNs alone struggle with understanding temporal dependencies and long-range relationships.

To address this, the Transformer model is integrated for its advanced attention mechanism, which captures global dependencies by evaluating the importance of different time steps in the sequence. This enhances the model's ability to grasp complex temporal relationships over extended periods. The BiLSTM networks are also included to further enhance temporal processing. BiLSTM improves upon traditional LSTMs by considering both past and future contexts, processing data in both forward and backward directions to capture dependencies across a broader temporal scope.

This synergistic combination leverages the CNN's spatial feature extraction, the Transformer's global attention, and the BiLSTM's bidirectional temporal processing, addressing the individual limitations of each component and optimizing the model's overall effectiveness in sewage flow forecasting.

The model first extracts multi-scale features through convolutional layers, followed by pooling layers for feature selection and simplification, emphasizing important features and reducing redundancy to enhance the model's correlation analysis capability. Utilizing Transformer layers to process one-dimensional convolution outputs, the model identifies and utilizes positional information in the time series. After deep learning of data sequence features by BiLSTM layers, the prediction results are output through fully connected layers, achieving precise extraction of temporal features.

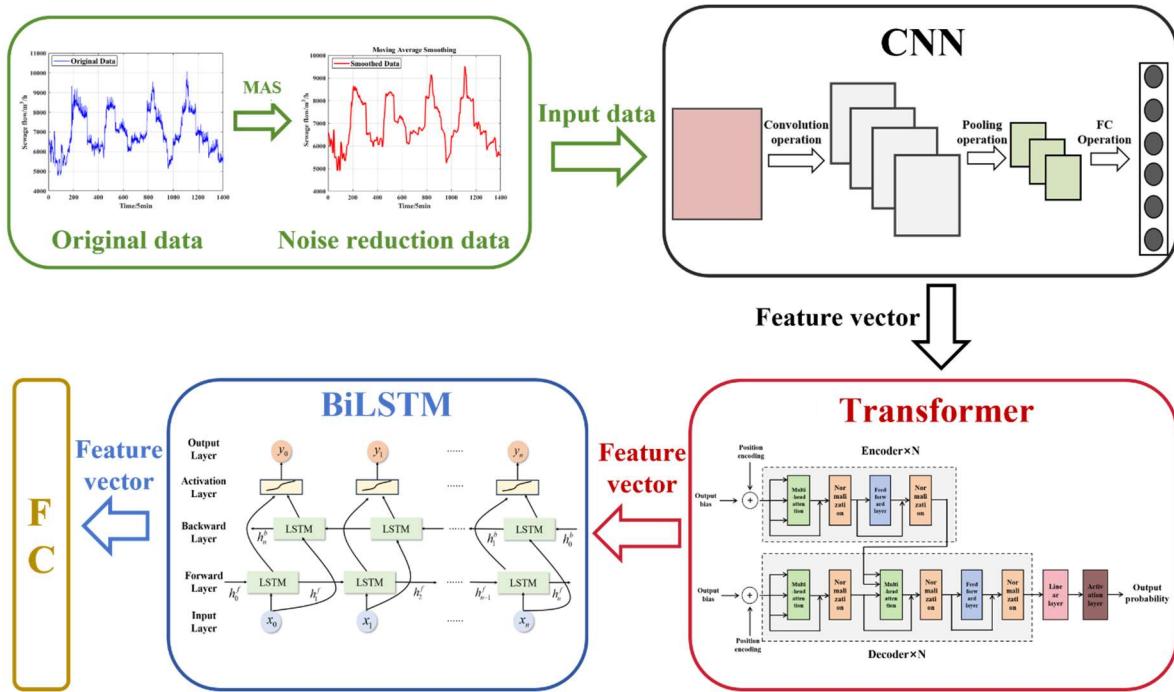


Figure 1. Model framework diagram for CNN-Transformer-BiLSTM.

2.2.1. CNNs

CNNs are a special type of feedforward neural network designed with convolution operations and deep architectures to address issues such as spatial information loss, low processing efficiency, and overfitting. With continuous improvements and optimizations in technology, CNNs have been widely applied in recent decades in areas such as image processing, fault diagnosis, and object recognition. CNN is mainly used to extract features from the data, so as to have a clearer understanding of the features of the data and facilitate subsequent prediction. The structure of CNN typically includes input layers, convolutional layers, pooling layers, fully connected layers, and output layers. Depending on the dimensionality of the data being processed, CNNs can be categorized into 1D convolution, 2D convolution, and 3D convolution. 1D convolution is used for processing time series data and language text, as illustrated in Figure 2.

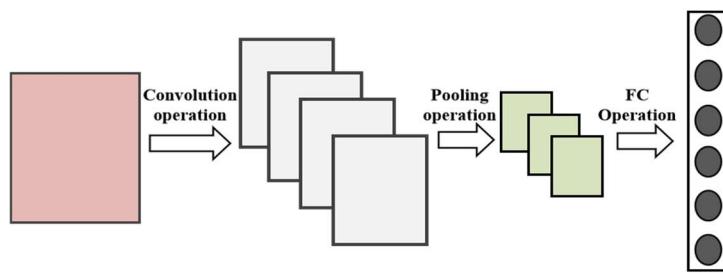


Figure 2. Model framework diagram for CNN.

(1) Convolutional layer

The convolutional layer is the core component of CNNs, responsible for extracting features from input data using specific convolution kernels. This layer performs convolution operations by sliding kernels over input data to generate feature maps. Unlike traditional fully connected layers, this design shifts to local connections and parameter sharing among kernels. Different convolution kernels are configured with distinct parameter sets, significantly reducing model complexity and the number of parameters. In 1D convolutional networks, operations are performed along a single dimension, adjusting channel numbers to achieve dimension transformation and feature extraction goals. The calculation process of 1D convolution is described as follows:

$$y_i = \sum_{i=1}^n w_i \cdot x_{t-i+1}, \quad (2)$$

where y_i denotes the output at a time step t , the number w_i represents the convolution kernel, and x_{t-i+1} is the input time series. Figure 3 illustrates the process where a 5×1 feature input is processed through a 3×1 convolutional kernel, resulting in an output. In this example, the input data is already processed by MAS, which spans from 07:05:00 on December 4, 2023 to 14:45:00 on January 12, 2024, and is convolved with the kernel to extract features. Additionally, another set of input data spanning from 00:00:00 on April 1, 2024 to 16:35:00 on May 7, 2024 is similarly processed.

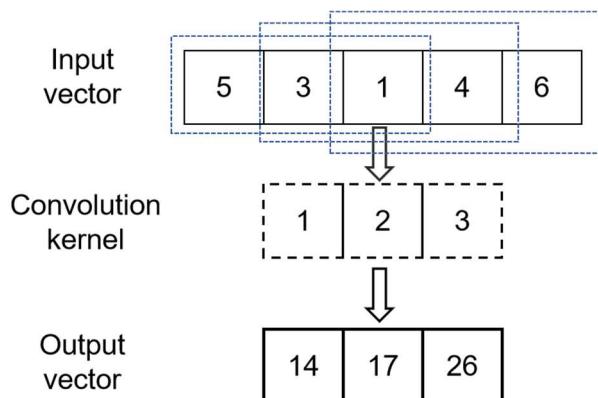


Figure 3. A process with one-dimensional convolution.

(2) Pooling layer

The pooling layer is typically positioned after the convolutional layer. Its main purpose is to use down sampling to lower the size of feature maps produced by the convolutional layer. This reduces computational flow and network parameters, albeit it may result in some feature information being

lost. Common pooling methods include max pooling and average pooling. In max pooling, the maximum value within each selected region serves as the representative value for that region, while average pooling computes the average of all values within the region to produce the output. An example process of max pooling is illustrated in Figure 4.

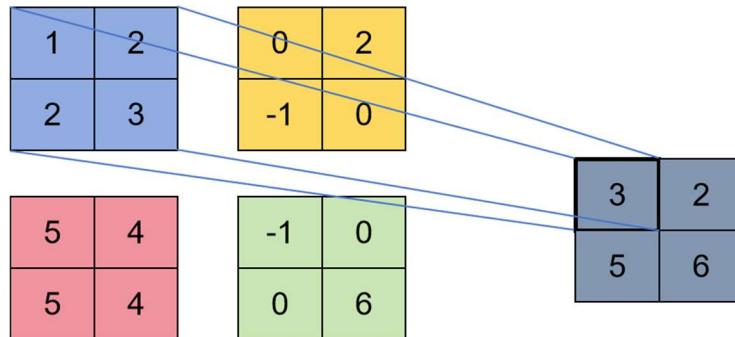


Figure 4. Max pooling operation.

(3) Fully connected layer

The fully connected layer follows the convolutional and pooling layers. Its role is to transform the output data from previous layers into a one-dimensional array, which serves as the input. It is the last phase of the model and is closely related to the earlier structures. It is in charge of the last data dimensionality modification and makes every effort to retain as much of the important information as possible. In CNNs, ReLU (rectified linear unit) activation functions are commonly applied after the pooling layer to enhance learning efficiency and processing capability.

2.2.2. Transformer feature extraction network

The transformer model gives weights to feature vectors to better predict sewage flow. The Transformer model first gained prominence in the field of machine translation, and its exceptional ability to capture complex relationships between positions in sequence data has garnered significant attention in academia. The core of this model consists of three main parts: an encoder, a decoder, and an attention mechanism.

The encoder includes an input layer, a positional encoding layer, and four layers of identical encoding units. The input layer is typically a fully connected layer that maps time-series data into a predefined dimensional space within the model. After the CNN layer performs feature extraction and dimensionality transformation on the input data, this processed data serves as the input to the Transformer model. The Transformer's input layer then projects this data into a higher-dimensional space suitable for the model's operations.

The positional encoding layer adds positional information to each element in the sequence, ensuring that the model can recognize the order of elements. This positional information allows the Transformer to understand the relative position of elements in the sequence, which is crucial for processing sequential data accurately.

The position-encoded sequence data is then fed into four encoding units. Each encoding unit processes the data using residual connections and layer normalization to maintain data integrity and regularity during transmission. The use of residual connections helps in preserving important information by allowing gradients to flow through the network without vanishing, while layer

normalization ensures stable and efficient training by normalizing the outputs of the previous layer.

The decoder predicts outputs based on the encoder's output and the previous decoder outputs at each time step, a process known as dynamic decoding. The decoder uses multi-head attention mechanisms to focus on different parts of the input sequence and the previously generated outputs, enabling it to generate accurate predictions by attending to various aspects of the data.

(1) Self-Attention mechanism

The attention mechanism, proposed by Mnih et al. [30] in 2014, aims to selectively focus on the most critical parts of vast amounts of information for intensive processing, while disregarding relatively unimportant information. By using this technique in sewage flow prediction models, the model can focus on the most important relationships between the input and output sequences, giving more weight to the relevant data in the prediction results and less weight to the less relevant data.

As a branch of the attention mechanism, the self-attention mechanism reduces the model's reliance on external information, focusing on exploring inherent connections between data or features. This mechanism demonstrates good performance in analyzing the characteristics of sewage flow and its influencing factors. By calculating weights within input vectors and adjusting them, it effectively captures interactions between preceding and subsequent elements in time-series data. The model architecture of the encoder combined with the self-attention mechanism is depicted in Figure 5, illustrating its structural layout for sewage flow prediction.

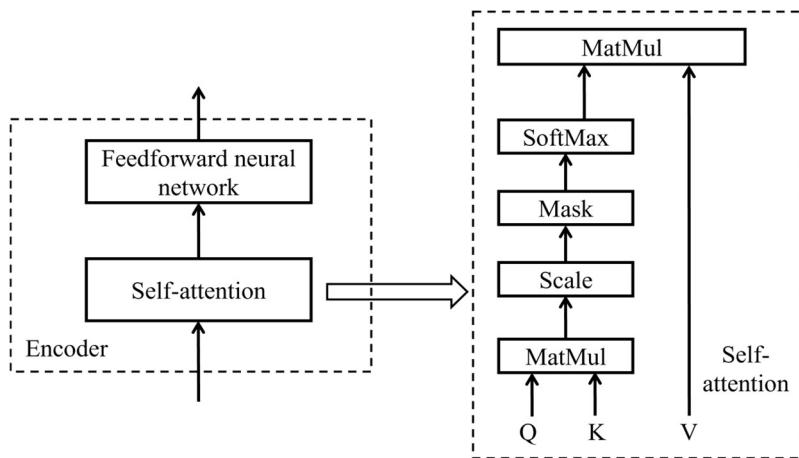


Figure 5. Encoder and self-attention mechanism architecture.

For each input vector X of the time series, it is mapped into three different spaces to generate query vector Q , key vector K , and value vector V . These are linearly mapped into three matrices w_Q , w_K , and w_V . The updated steps are as follows:

- 1) Generate query vectors:

$$\begin{cases} Q = X \cdot W_Q, \\ K = X \cdot W_K, \\ V = X \cdot W_V. \end{cases} \quad (3)$$

- 2) Use scaled dot-product as the attention scoring function:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{D_k}}\right) \cdot V, \quad (4)$$

where $Q \cdot K^T$ represents the scores for each sample in the vector, and $\sqrt{D_k}$ represents the training gradients that need to be optimized.

- 3) Normalize the residual network and use the result as the input of the feedforward neural network.
- (2) Multi-head attention mechanism

The multi-head attention mechanism [31] is built on several self-attention units, which are parallelly combined. Each self-attention unit is also called a ‘head’, and each head focuses on different information dimensions. Thus, by combining multiple self-attention heads that focus on their respective priorities, the multi-head attention mechanism can achieve more extensive and in-depth information captured when dealing with complex tasks. Each head operates independently without interference from each other. Figure 6 illustrates the architecture of the multi-head attention mechanism.

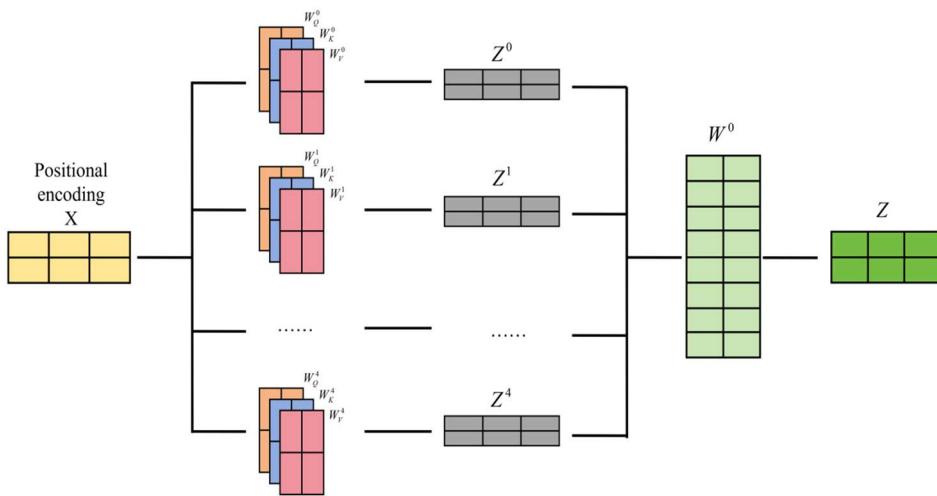


Figure 6. Structure of the multiple attention mechanism.

The concatenation formulas for the multi-head attention mechanism are:

$$\text{Multi}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2) \cdot W^0, \quad (5)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \quad (6)$$

The basic structure of the Transformer is shown in Figure 7. The left part illustrates the encoder structure, where each layer consists of two sub-layers: the first sub-layer is composed of the multi-head attention mechanism, residual connections, and layer normalization, and the second sub-layer is composed of the feedforward fully connected networks (FFN), residual connections, and layer normalization.

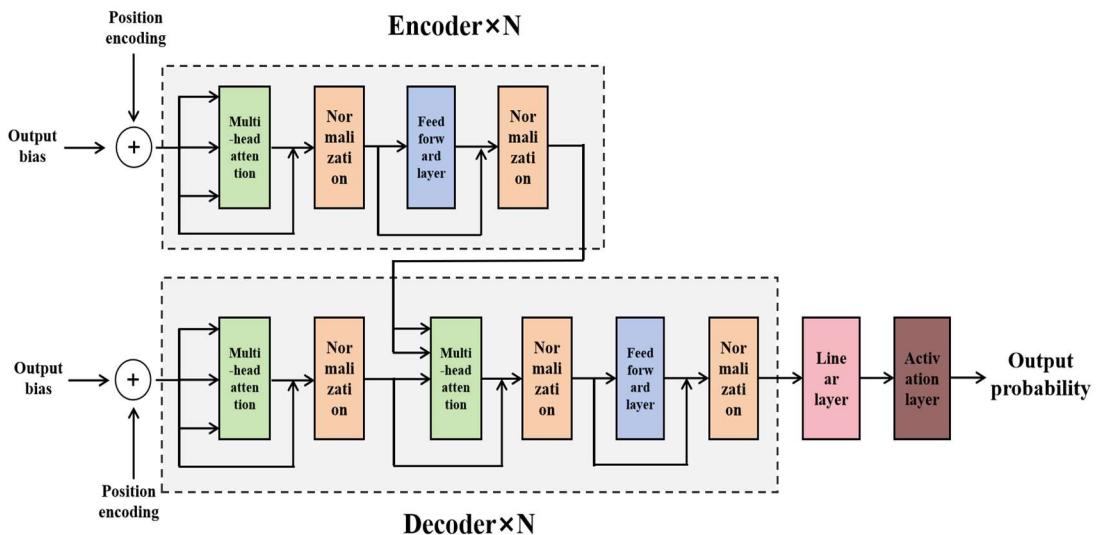


Figure 7. The basic structure of Transformer.

2.2.3. BiLSTM

LSTM neural networks are a type of RNN (recurrent neural network) specifically designed to address the issue of long-term dependencies that traditional RNNs struggle with. All RNNs have a chain-like structure of repeating neural network modules. In standard RNNs, this repeating module has a very simple structure, such as a tanh layer. LSTM neural networks replace the simple hidden layer neurons of RNNs with gate mechanisms, which can effectively tackle long-term dependency issues and perform exceptionally well on sequential tasks. LSTM neural networks exhibit excellent nonlinear fitting capabilities and have become one of the most popular frameworks in deep learning. They resolve the issue of insufficient cognitive abilities in shallow networks. The specific principles for LSTM are in the attachment.

Traditional LSTM networks can only encode forward based on historical states and cannot consider the influence of backward sequences. However, changes in sewage flow data are closely related to time development, where future data typically resemble past data. To achieve more comprehensive and accurate predictions, it is necessary to consider the impact of backward sequences. BiLSTM networks introduce the concept of bidirectional computation. They enable simultaneous forward and backward computations based on the original LSTM network. BiLSTM networks can extract both forward and backward information, better exploring the temporal characteristics of sewage flow data and further improving the accuracy of prediction models. BiLSTM has significant advantages in wastewater flow prediction. Through a bidirectional processing mechanism, BiLSTM is able to capture both forward and backward information in time series. Its long short-term memory (SM) allows BiLSTM to effectively deal with long-term dependency problems and capture seasonal changes and long-term trends in traffic data. This capability of BiLSTM allows it to provide more accurate and stable results when predicting wastewater flows. By integrating historical data, BiLSTM not only takes into account past traffic trends, but also incorporates future information to improve prediction accuracy.

The data processed by the Transformer layer is compiled and decoded, and the resulting feature-extracted data is then used as input for the BiLSTM. During the forward pass, the input at current time t is composed of external input data and the output value of the previous unit at time $t-1$. In contrast, during the backward pass, the input at time t requires the output value of the state at time $t+1$ as the

output of the previous unit differs from the forward pass. The structure diagram of the BiLSTM network is shown in Figure 8.

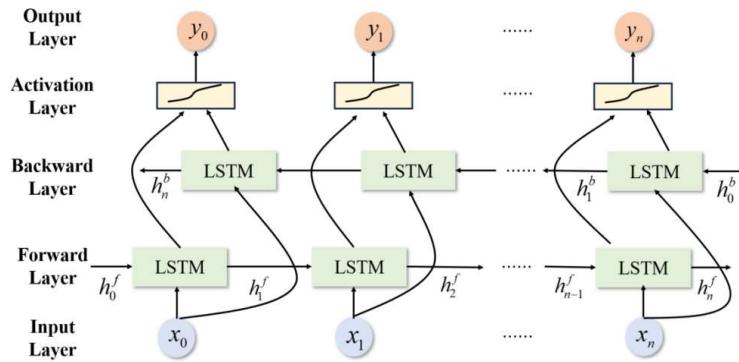


Figure 8. Model framework diagram for BiLSTM.

The final output of the BiLSTM network is determined by both the forward and backward outputs. Therefore, the inputs and outputs of the forward and backward passes are different, as shown in Eqs (7) to (10):

$$\overset{1}{h}_t = f(W_1 x_t + U_1 \overset{1}{h}_{t-1} + b_1), \quad (7)$$

$$\overset{2}{h}_t = f(W_2 x_t + U_2 \overset{2}{h}_{t-1} + b_2), \quad (8)$$

$$\overset{1}{h}_t = \overset{1}{h}_t \oplus \overset{2}{h}_t, \quad (9)$$

$$O_t = g(V \overset{1}{h}_t + b_3), \quad (10)$$

where $\overset{1}{h}_t$ denotes the forward output of the LSTM network, W_1 represents the input layer weight matrix for forward propagation, U_1 represents the hidden layer weight matrix, and b_1 represents the bias vector. $\overset{2}{h}_t$ denotes the backward output of the LSTM network, W_2 represents the input layer weight matrix for backward propagation, U_2 represents the hidden layer weight matrix, and b_2 represents the bias vector. O_t represents the final output value, V represents the output layer weight matrix, b_3 represents the bias vector, and \oplus denotes concatenation.

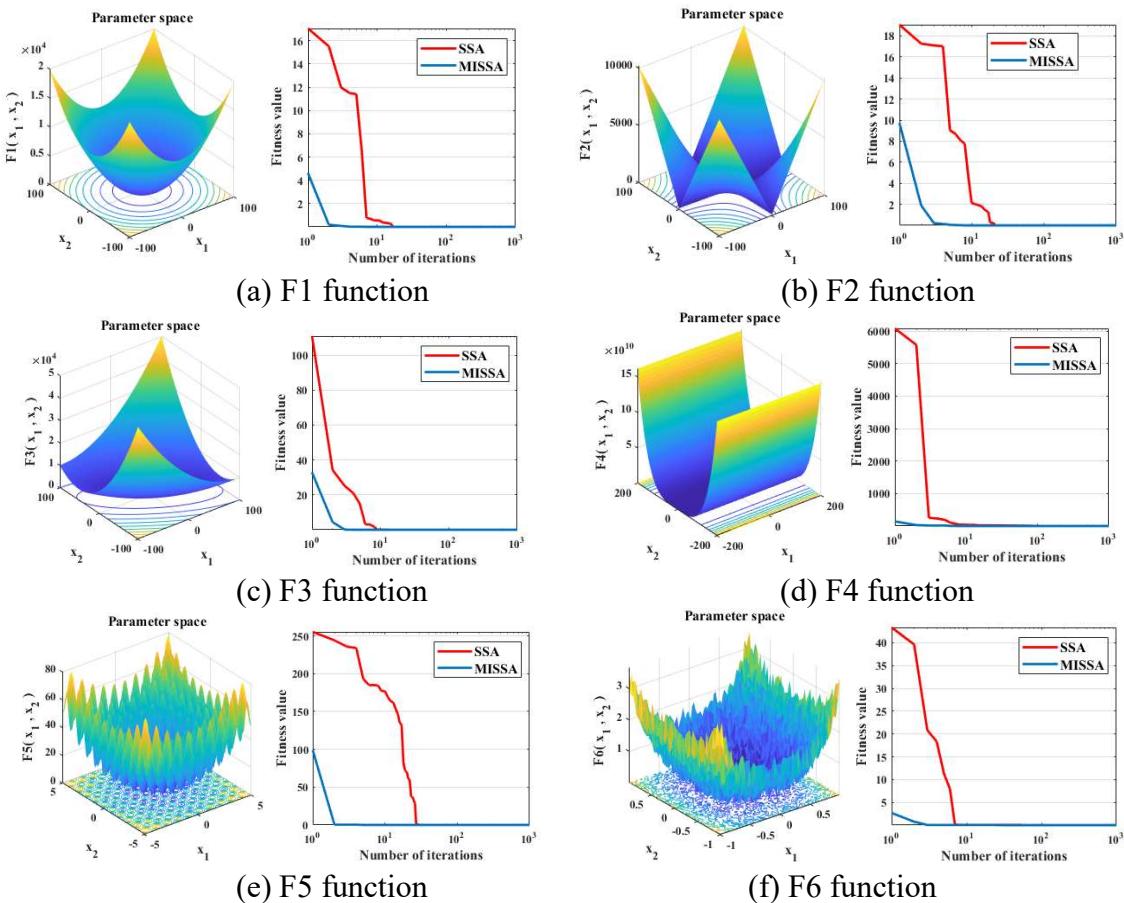
2.3. An improved SSA

In this study, six benchmark test functions with unimodal and multimodal characteristics, all having optimal solutions of 0, were selected to test the performance of SSA and proposed an improved SSA. The basic information of the selected benchmark test functions was shown in Table 1. The first four functions are unimodal, and the last two are multimodal functions. Additionally, the parameters for both an improved SSA and SSA algorithms were set as follows: The initial population size was set to 20, the maximum number of iterations was set to 1000, the warning value was set to 0.6, the proportion of discoverers was set to 0.7, and the proportion of the population perceiving danger was set to 0.2. To ensure the fairness and effectiveness of the experiments, all tests were conducted in the same operating environment.

Table 1. Benchmark functions.

Function	Expression	Dimension	Range	Optimal value
Sphere (F1)	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
Schwefel 2.22 (F2)	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
Schwefel 1.2 (F3)	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100,100]	0
Rosenbrock (F4)	$F_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-100,100]	0
Griewank (F5)	$F_5(x) = \frac{1}{400} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-5.12,5.12]	0
Rastrigin (F6)	$F_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-600,600]	0

Figure 9 shows the convergence graphs of the algorithm optimization benchmark test functions. From the convergence curves of an improved SSA and SSA, it can be observed that an improved SSA's convergence speed is significantly faster than SSA's for both unimodal and multimodal test functions, requiring 10–25% less iterations to find the optimal solution of the function.

**Figure 9.** Algorithm optimization benchmark function test results.

This indicates that an improved SSA has a stronger optimization capability. In the multimodal test functions, an improved SSA's convergence curves are consistently lower than those of SSA, demonstrating that an improved SSA algorithm has higher convergence precision and stronger global search ability. By comparing the six test functions, it can be concluded that an improved SSA algorithm is characterized by high search efficiency and strong optimization capability, while also verifying the effectiveness of the proposed improvement strategies.

3. Multi-strategy integrated improved SSA

The selection of hyperparameters for predictive models can be viewed as an optimization problem, typically addressed using precise algorithms such as Bayesian optimization, gradient descent, and stochastic gradient descent. While precise algorithms can obtain exact solutions to optimization problems, they often exhibit lower efficiency in computation. In contrast, heuristic algorithms possess powerful optimization capabilities and operate with high efficiency, making them competitive in solving hyperparameter optimization problems. Therefore, this paper proposes an intelligent optimization algorithm called an improved SSA to address the issue of hyperparameter optimization.

3.1. Traditional SSA

The core idea of the SSA [32] is to mimic the foraging and antipredatory behaviors of sparrows, applying them to local and global search optimization processes. The specific principles for traditional SSA are in the attachment.

The SSA has advantages such as high search efficiency and simple parameter settings, but it also has the following shortcomings.

(1) The random generation of the initial population leads to uneven distribution within the population, reducing diversity and resulting in poorer initial solutions, thereby affecting the convergence speed of the algorithm.

(2) In the early stages of population search, discoverers may approach the global optimum easily. However, due to limited exploration space, they may get trapped in local optima, making it difficult to escape.

(3) The control parameters β and k for step size are crucial for balancing global search and local exploitation capabilities. However, being random variables, their uncertainty makes it challenging to effectively coordinate global search and local exploitation, potentially causing the algorithm to converge to local optima.

(4) As the algorithm progresses through iterations, the reduction in population diversity may lead to a decrease in the quality of the optimal solution and weaker global search capability. Therefore, there is still room for further optimization of the SSA to enhance its optimization capabilities.

3.2. Multiple optimization strategies

To enhance the optimization accuracy of the SSA algorithm, this paper adopts a series of strategies.

(1) Utilization of Sin chaotic mapping mechanism [33] for population initialization: This strategy aims to improve population diversity and the quality of initial solutions.

(2) Dynamic adaptive weight improvement for discoverer position updates: This approach enhances the algorithm's convergence speed and search efficiency.

(3) Introduction of Cauchy mutation mechanism [34]: This mechanism is introduced to strengthen the algorithm's global search capability, aiding in escaping local optima.

(4) Implementation of random reverse learning strategy [35]: This strategy is employed to obtain corresponding reverse solutions, thereby enhancing the algorithm's optimization efficiency and stability.

3.2.1. Chaotic mapping mechanism

Chaos variables are commonly used in optimization problems due to their traversal and regularity properties. Logistic and Tent mappings are well-known chaotic models; however, they have limited folding times within the iteration region and exhibit a significant number of rational fixed points, adversely affecting mapping quality. In contrast, the Sin mapping is a chaotic model with an infinite folding number, characterized by uniform traversal and rapid convergence advantages. Therefore, this paper chooses to use the Sin mapping for initializing the SSA population. The one-dimensional self-mapping expression of Sin chaotic mapping is as follows:

$$\begin{cases} z_{n+1} = \sin\left(\frac{\alpha\pi}{z_n}\right), \alpha \in (0,1), \\ -1 \leq z_n \leq 1, z_n \neq 0. \end{cases} \quad (11)$$

Mapping the Sin chaotic mapping into the search space yields the initial positions of the population:

$$x_i = x_{lb} + (x_{ub} - x_{lb}) \frac{1+z_i}{2}, \quad (12)$$

where x_{lb} and x_{ub} represent the lower and upper bounds of each individual in each dimension, and z_i denotes the chaotic sequence generated by Eq (11).

3.2.2. Dynamic adaptive weight

To prevent the algorithm from falling into local optima, this paper proposes an improved way to update the discoverer's position, incorporating the global best solution from the previous generation. Thus, the update of the discoverer's position is influenced by historical optimal solutions, thereby increasing the algorithm's ability to escape local optima. Meanwhile, a dynamic weight factor ω is introduced to balance the algorithm's global search capability and local exploration capability. Initially, ω has a larger value in the early iterations, facilitating global exploration. In later iterations, it adaptively decreases, promoting local search and enhancing convergence speed. The formula for calculating the weight coefficient ω and the updated discoverer's position after improvement are as follows:

$$\omega = \frac{k}{e^{-(1-\frac{t}{T_{max}})}}, k \in [1, 2], \quad (13)$$

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + \omega(f_{g,j}^t - f_{i,j}^t) \cdot rand, & R < ST, \\ X_{i,j}^t + Q, & R \geq ST, \end{cases} \quad (14)$$

where $f_{g,j}^t$ is the j_{th} dimensional globally optimal solution in the previous generation.

3.2.3. Improved scout position update

The parameters β and k , controlling the step size, are crucial for balancing global search capability and local exploitation capability. However, due to their random nature, they introduce significant uncertainty, which may lead to uncontrolled updates of scout positions, causing them to deviate from the global optimal position. Therefore, this paper proposes an improved method for updating scout positions:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t|, & f_i \neq f_g, \\ X_{best}^t + \beta \cdot |X_{worst}^t - X_{best}^t|, & f_i = f_g. \end{cases} \quad (15)$$

The improved position update formula indicates that if a sparrow individual is at the current best position, it will fly toward a randomly chosen position between the best and worst positions. Otherwise, it will choose to fly toward a randomly chosen position between its current position and the best position.

3.2.4. Fusion of Cauchy mutation and random backward learning strategy

Backward learning is an important strategy that improves algorithmic optimization by using the current solution as a basis to search for corresponding backward solutions, keeping the better solution after evaluation and comparison. However, traditional backward learning strategies generate backward solutions with a fixed distance from the current solution, lacking sufficient randomness. This limitation hinders effective diversification of the population and restricts exploration of the search space. To address this issue, this paper proposes a random backward learning strategy, introducing a random factor to enhance the position of backward solutions. This enhances the algorithm's ability to escape local optima and increases population diversity. The formula for random backward learning is as follows:

$$X_{best}^{t''} = l_1 ub + l_2 (lb - X_{best}^t), \quad (16)$$

$$X_{i,j}^{t+1} = X_{best}^{t''} + b_1 (X_{best}^t - X_{best}^{t''}), \quad (17)$$

where $X_{best}^{t''}$ is the backward solution of the best solution found in the t_{th} iteration, l_1 and l_2 are the lower and upper bounds, and b_1 represents control parameters for information exchange. The formula is as follows:

$$b_1 = (T_{max} - \frac{t}{T_{max}})^t. \quad (18)$$

During foraging, joiners gather around the discoverer and may engage in food competition, potentially allowing a joiner to replace the discoverer and exacerbate the algorithm's tendency to fall into local optima. To prevent this scenario, this paper introduces a Cauchy mutation strategy [63] to enhance the update formula of the joiners, thereby improving the algorithm's global optimization capability. The new joiner position update formula is as follows:

$$X_{i,j}^{t+1} = X_{best}^t + cauchy(0,1) \cdot X_{best}^t, \quad (19)$$

where $cauchy(0,1)$ follows a standard Cauchy distribution. The Cauchy distribution random variable generation function is $\eta = \tan [(\xi - 0.5) \pi]$. To further enhance the algorithm's optimization

performance, a dynamic selection strategy is employed to update the target position. This strategy involves alternating between random backward learning and Cauchy mutation disturbance methods with a certain probability, dynamically updating the target position. The selection state of these two improvement strategies is determined by a selection probability, calculated as follows:

$$P_s = -e^{(1-\frac{t}{T_{max}})^{10}}. \quad (20)$$

Specifically, the selection process is as follows: if $\text{rand} < P_s$, select Eqs (16) to (18) for position updates based on the random backward learning strategy. Otherwise, choose Eq (19) for updating the target position based on the Cauchy mutation disturbance strategy, enhancing the algorithm's ability to escape local spaces. The pseudo-code for the proposed and improved SSA in this paper is shown below:

Algorithm: Pseudo-code for an improved SSA

- 1: Parameter initialization: Maximum iterations T_{\max} , number of discoverers PD , percentage m of joiners, warning value R
- 2: Population initialization: Initialize sparrow population based on Eqs (22) and (23) using Sin mapping
- 3: Compute fitness value for each sparrow based on Eq (18)
- 4: **Loop**
- 5: while $t < T_{\max}$
- 6: Sort fitness values to find the current best and worst individuals
- 7: Set $R = \text{rand}(1)$
- 8: **for** $i = 1$ to PD **do**
- 9: Update discoverer positions based on Eq (25)
- 10: **end for**
- 11: **for** $i = (1+PD)$ to n **do**
- 12: Update joiner positions based on Eq (30)
- 13: **end for**
- 14: **for** $i = 1$ to $(m \times n)$ **do**
- 15: Update scout positions based on Eq (26)
- 16: **end for**
- 17: Perturb the current best solution using an appropriate strategy based on Eq (31) to generate a new solution
- 18: Check termination condition. If met, proceed; otherwise, go to step 6
- 19: **t=t+1**
- 20: **End loop**
- 21: Output results

3.3. Algorithm optimization steps

The proposed and improved SSA algorithm is applied to hyperparameter optimization of the CNN-Transformer-BiLSTM model. The model framework is divided into the data preprocessing module, an improved SSA optimization module, and CNN-Transformer-BiLSTM feature extraction module.

The data preprocessing module performs noise reduction operations on collected data and divides it into training and testing sets. The an improved SSA optimization module controls the movement of scouts, discoverers, and joiners based on fitness values, iterating to optimize global solutions and population structure. This method ultimately obtains optimized hyperparameters and network models. The CNN-Transformer-BiLSTM feature extraction module decodes the hyperparameters optimized by an improved SSA, retrieving iteration counts, learning rates, and hidden layer node numbers. Next, using the training set, the model is trained and predictions are made on the testing set to obtain actual values, predicted values, and their errors. Fitness values are computed using root mean square error (RMSE) and fed back into an improved SSA optimization module until convergence of the loss function. The overall model framework is illustrated in Figure 10.

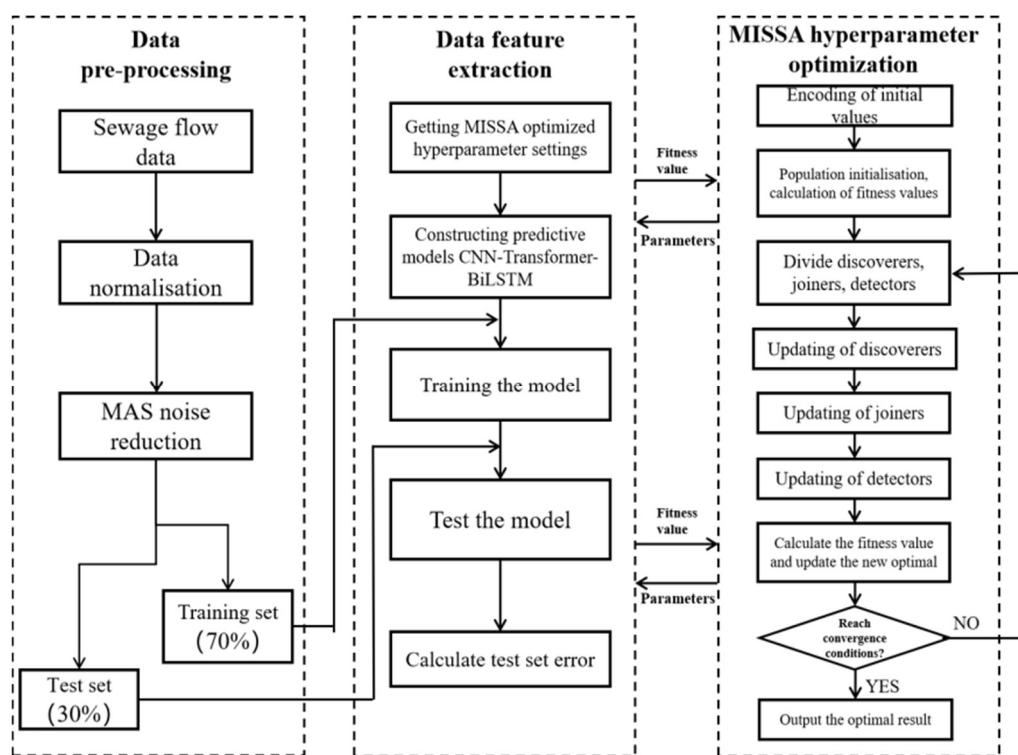


Figure 10. The framework diagram of an improved SSA-CNN-Transformer-BiLSTM.

4. Results and analysis

4.1. Simulation environment

(1) Dataset description

The experimental data for this study is the monitored sewage flow of the influent of a wastewater treatment plant in a city of Guangdong Province, China. The dataset includes historical sewage flow data and corresponding instantaneous rainfall data. We chose two datasets according to the

corresponding instantaneous rainfall: the first dataset spans from 07:05:00 on December 4, 2023, to 14:45:00 on January 12, 2024, and the second one spans from 00:00:00 on April 1, 2024, to 16:35:00 on May 7, 2024. In the first dataset, the range of sewage flow is from 3252.502 to 10137.42 cubic meters per hour, and the average sewage flow is 7480.499 cubic meters per hour. There are no rainfall events at all. In the second dataset, the range of sewage flow is from 3399.66 to 10406.79 cubic meters per hour, and the average sewage flow is 7356.69 cubic meters per hour. There are 42 rainfall events: the average total rainfall of the 42 rainfall events are 10.07 millimeters; the maximum total rainfall of the 42 rainfall events are 66.40 millimeters; the minimum total rainfall of the 42 rainfall events are 0.20 millimeters. We chose the datasets this way in order to test our model under different rainfall and temperature conditions. Data sampling was performed at 5-minute intervals, resulting in 288 sampling points per day. The first dataset used for model training comprises the first 9544 samples, while the last 1024 data samples are used for testing the model's predictive output. The second dataset used for model training comprises the first 8520 samples, while the last 2048 data samples are used for testing the model's predictive output.

(2) Experimental environment and parameter settings

The experiments were conducted on a Windows 10 system, using PyCharm as the integrated development environment. The experiments were performed using Python 3.9, with an Intel Core i5-12100F CPU, an NVIDIA GeForce RTX 3060 GPU, and 32GB of RAM. The sliding window size was set to 7. The an improved SSA algorithm was used for model parameter optimization, with each model training iteration set to 10 iterations. By integrating prior knowledge, domain expertise, and relevant literature, this study has identified the optimal hyperparameter settings for an improved SSA-CNN-Transformer-BiLSTM model, as shown in Table 2.

Table 2. Parameter settings for CNN-Transformer-BiLSTM network.

Network name	Hyperparameter type	Parameter optimization range
CNN	Number of convolutional layers	[1, 5]
BiLSTM	Number of layers	[1, 5]
	Number of neurons	[16, 256]
	Number of attention heads	(2, 3, 5, 6, 10, 15)
Transformer	Number of encoder layers	[1, 5]
	Number of encoder hidden layer dimensions	[16, 256]
Adam	Initial learning rate	[0.00001, 0.01]
	Epoch	20

The parameter settings of an improved SSA algorithm are shown in Table 3:

Table 3. Parameter settings of an improved SSA algorithm.

Parameter	Set value
Initial population size	20
Maximum number of iterations	10
Early warning value	0.6
Proportion of discoverers	0.7
Proportion of sparrows sensing danger	0.2

(3) Evaluation metrics

To evaluate the effectiveness and applicability of the prediction model, it is necessary to design reasonable evaluation metrics for performance assessment. Commonly used prediction error evaluation metrics [36] include *RMSE*, mean absolute error (*MAE*), and the coefficient of determination (R^2). Their expressions are as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}, \quad (21)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (22)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2}, \quad (23)$$

where n is the number of samples, y_i is the monitored value of sample i , \bar{y} is the mean of the monitored values, and \hat{y}_i is the predicted value of sample i . Among these metrics, the closer the values of *RMSE* and *MAE* are to 0, the better performance of the model, while the closer R^2 is to 1, the better performance of the model.

4.2. Model validation

The loss function variation curve of the CNN-Transformer-BiLSTM prediction model during the training process is shown in Figures 11 and 12. It can be observed that the loss function values in both the training and testing sets gradually decrease and approach zero. The closer the loss function value is to zero, the better the training result of the prediction model. This also indicates that the prediction model can find optimal hyperparameter values within a limited number of iterations.

The loss function formula is as follows:

$$J = \frac{1}{m} \sum_{i=1}^m (\bar{y}_i - y_i)^2, \quad (24)$$

where m is the number of samples, y_i is the monitored value of sample i , and \bar{y}_i is the predicted value of sample i .

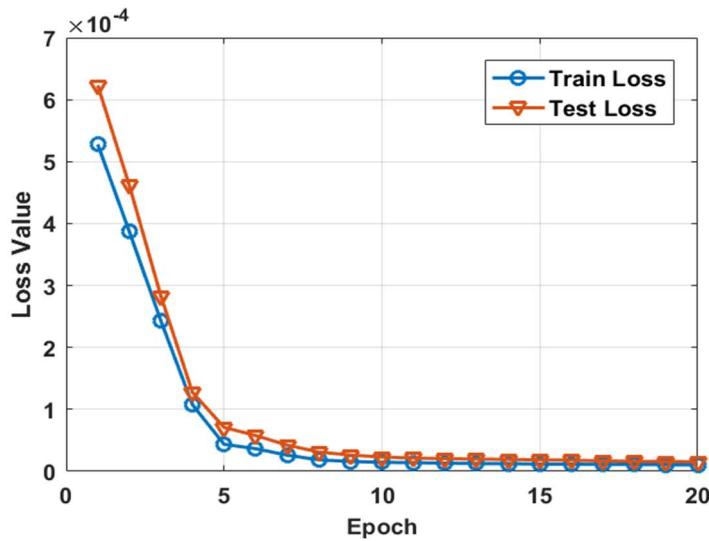


Figure 11. Loss function variation curve.(07:05:00 on Dec.4, 2023-14:45:00 on Jan.12, 2024.).

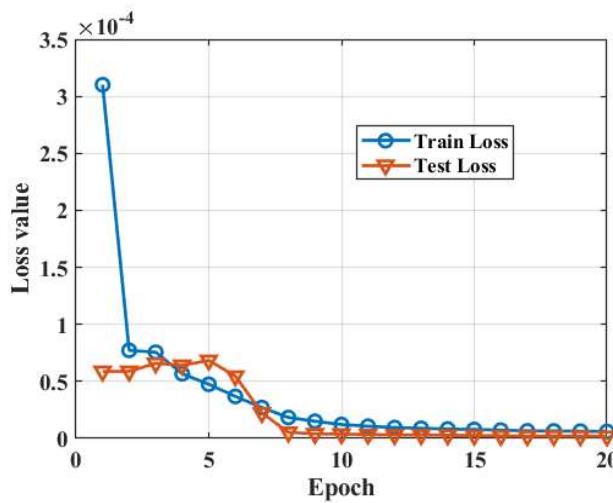


Figure 12. Loss function variation curve.(00:00:00 on Apr.1, 2024-16:35:00 on May.7, 2024.).

To validate the predictive capability and performance of the CNN-Transformer-BiLSTM model constructed in this chapter, the predicted sewage flow from the CNN-Transformer-BiLSTM model is compared with the monitored values. The comparison results are shown in Figures 13 and 14.

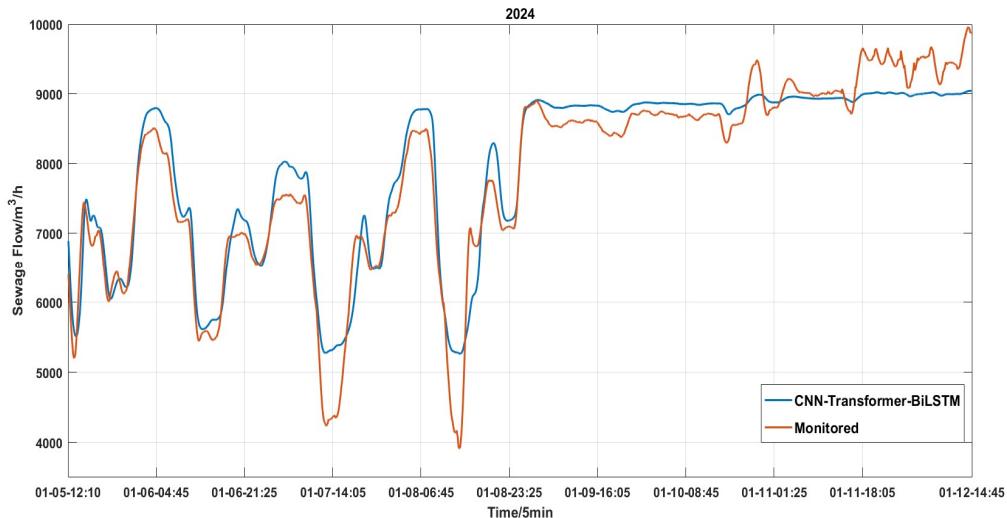


Figure 13. Comparison between predicted flow from CNN-Transformer-BiLSTM model and monitored sewage flow (1).

As shown in Figure 13, the predicted sewage flow of the CNN-Transformer-BiLSTM model is close to the monitored flow values with R^2 of 0.92115, indicating that the proposed CNN-Transformer-BiGRU model can effectively capture the trend of sewage flow changes [50]. The difference between the predicted peak/trough flow and monitored peak/trough flow is within 13.5%, and the predicted peak/trough flow appearance time is within 2.5 minutes away from the monitored peak/trough flow time.

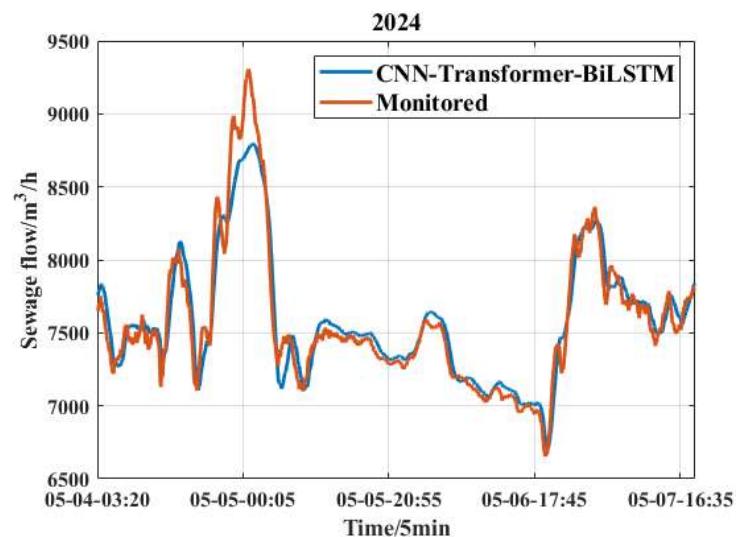


Figure 14. Comparison between predicted flow from CNN-Transformer-BiLSTM model and monitored sewage flow (2).

As shown in Figure 14, the predicted sewage flow of the CNN-Transformer-BiLSTM model is close to the monitored flow values with R^2 of 0.85328, indicating that the proposed CNN-Transformer-BiGRU model can effectively capture the trend of sewage flow changes. The difference between the predicted peak/trough flow and monitored peak/trough flow is within 3.6%, and the predicted peak/trough flow appearance time is within 2.5 minutes away from the monitored peak/trough flow time.

From the above two graphs and analysis, it can be seen that in January, when there was no precipitation, the R^2 reaches 0.92115. In May, when there was precipitation, the R^2 decrease to 0.85328. Although the R^2 decreases, the prediction of the predicted peak/trough flow and monitored peak/trough flow is relatively accurate. It can be seen that although the accuracy of the model is slightly weakened when there are rainfall events, the grasp and accuracy of the overall trend are still very good, indicating that our model can be well applied under different rainfall and temperature conditions.

4.3. Ablation experiment

In this section, we conduct ablation experiments to verify the contribution of each component to the overall performance of the model. This research method involves gradually adding key components (CNN, Transformer) to the model and observing their specific impact on model performance. In the first experiments, the first 9544 samples are used for model training, while the last 1024 data samples are used for testing the model's predictive output. In the second experiments, the first 8520 samples are used for model training, while the last 2048 data samples are used for testing the model's predictive output. The experiments are repeated five times, with the best and worst prediction results removed. The average sewage flow predictions from the remaining three experiments are recorded and used to plot the model's flow prediction curve, as shown in Figures 15 and 16.

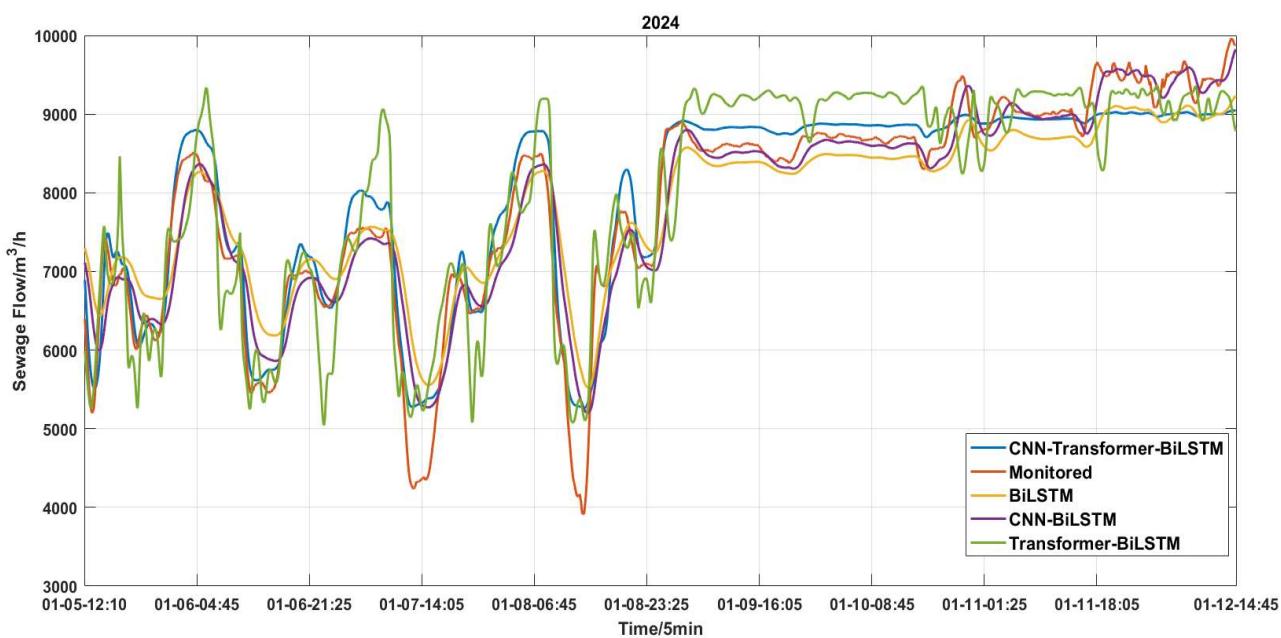


Figure 15. Predicted flow curves of different models versus monitored flow (1).

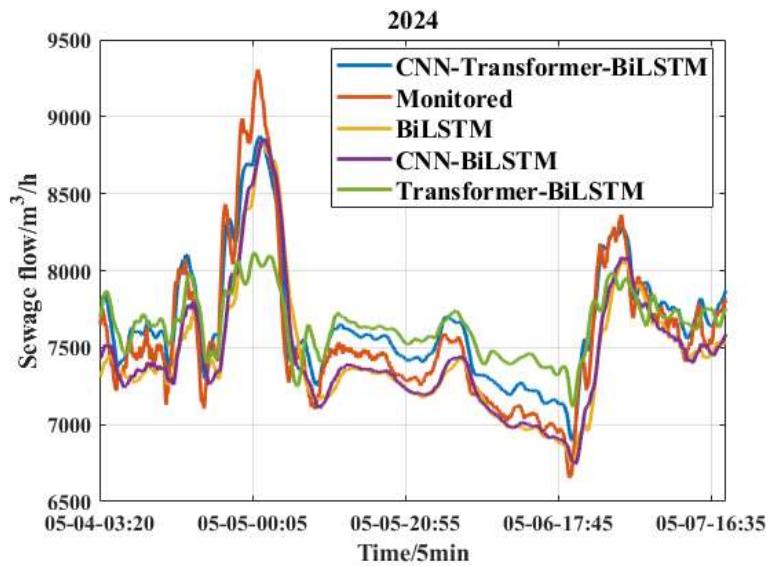


Figure 16. Predicted flow curves of different models versus monitored flow (2).

As observed in Figures 15 and 16, the CNN-Transformer-BiGRU (bidirectional gated recurrent unit) model fits the monitored sewage flow trend better than other models. To more comprehensively and scientifically evaluate the performance of each model, the evaluation metrics of the models were calculated and the results are shown in Tables 4 and 5.

Table 4. Results of evaluation metrics for different forecasting models. (match Figure 15).

Model name	R^2	RMSE	MAE
BiLSTM	0.83048	558.9778	420.1193
CNN-BiLSTM	0.89421	441.58713	292.36119
Transformer-BiLSTM	0.82317	570.91810	468.01435
CNN-Transformer-BiLSTM	0.92115	381.23022	300.22074

As shown in Table 4, the basic BiLSTM model performed moderately, with an R^2 of 0.83048, RMSE of 558.9778, and MAE of 420.1193. This indicates that using the BiLSTM model alone cannot fully capture the complex features of the data, leaving significant room for improvement.

After adding the CNN module, the R^2 increased to 0.89421, while the RMSE and MAE decreased to 441.58713 and 292.36119, respectively. This shows that the CNN module has significant advantages in capturing local features and can extract more effective information, thereby improving the overall performance of the model.

However, when the Transformer module was added to the BiLSTM model, the model performance didn't improve further, with R^2 decreasing to 0.82317, and RMSE and MAE increasing. This indicates that the Transformer module has problems in handling long-sequence dependencies, and it may be due to the separation with CNN.

Finally, the model combining both the CNN and Transformer modules performed the best, with R^2 reaching 0.92115, RMSE decreasing to 318.23022, and MAE to 300.22074. This demonstrates the powerful capability of the combined CNN and Transformer modules in capturing both local features and global dependencies, significantly enhancing the overall performance of the model and validating the applicability and superiority of incorporating the CNN and Transformer networks.

Table 5. Results of evaluation metrics for different forecasting models. (match Figure 16).

Model name	R^2	RMSE	MAE
BiLSTM	0.73859	245.5582	180.7114
CNN-BiLSTM	0.81971	203.9268	137.8229
Transformer-BiLSTM	0.72807	265.1825	184.7853
CNN-Transformer-BiLSTM	0.85328	183.9647	145.3357

As shown in Table 5, the basic BiLSTM model performed moderately, with an R^2 of 0.73859, RMSE of 245.5582, and MAE of 180.7114. This indicates that using the BiLSTM model alone cannot fully capture the complex features of the data, leaving significant room for improvement.

After adding the CNN module, the R^2 increased to 0.81971, while the RMSE and MAE decreased to 203.9268 and 137.8229, respectively. This shows that the CNN module has significant advantages in capturing local features and can extract more effective information, thereby improving the overall performance of the model.

However, when the Transformer module was added to the BiLSTM model, the model performance didn't improve further, with R^2 decreasing to 0.72807, and RMSE and MAE increasing. This indicates that the Transformer module has problems in handling long-sequence dependencies, and it may be due to the separation with CNN.

Finally, the model combining both the CNN and Transformer modules performed the best, with R^2 reaching 0.85328, RMSE decreasing to 183.96479, and MAE to 145.3357. This demonstrates the powerful capability of the combined CNN and Transformer modules in capturing both local features and global dependencies, significantly enhancing the overall performance of the model and validating the applicability and superiority of incorporating the CNN and Transformer networks.

4.4. Algorithm validation

The purpose of this section is to verify the effectiveness and superiority of an improved SSA algorithm by comparing its performance against some other SSA variant algorithms and newer metaheuristic algorithms on a competition dataset. The selected comparison metaheuristic algorithms are as follows: improved SSA (ISSA) - [38], improved PSO algorithm (ISPSO) - [39], adaptive genetic algorithm (AGA) - [40], grey wolf algorithm (GWO) - [41], whale optimization algorithm (WOA) - [42], dung beetle optimization algorithm (DBO) - [43], and african vultures optimization algorithm (AVOA) - [44]. Additionally, appropriate parameters that significantly reflect the performance of these algorithms were used, as detailed in Table 6. In this experiment, the dimension was set to 20, the maximum number of iterations was set to 1000, and the population size was set to 20. The experimental results are based on 2000 runs of the algorithms.

Table 6. Parameter settings of different algorithms in the experiments.

Algorithm	Parameter	Value	Algorithm	Parameter	Value
GWO	$\alpha_{max}, \alpha_{min}$	2, 0	AGA	P_c, P_d	0.5, 0.05
ISSA	PD, SD, R	0.7, 0.2, 0.6	ISPSO	$\omega_{max}, \omega_{min}, c, m$	0.9, 0.2, 1.496, 5
DBO	k, b	0.1, 0.3	AVOA	$\omega_1, \omega_2, c_1, r_1$	0.2, 0.2, 0.5, 2
WOA	a, b, ρ	1, 1, 0.5	An improved SSA	PD, m, R	0.7, 0.2, 0.6

To evaluate the comprehensive performance of the above algorithms, we selected some of the most challenging publicly available test functions from the CEC (congress on evolutionary computation) 2017 test suite [45]. These functions include 3 single-modal and 7 multimodal test functions. Details of the test functions can be found in Table 7, and all problems are minimization problems.

Table 7. CEC2017- unimodal as well as multimodal test functions.

Function modality	No.	Function name	Range of values	Optimal value
Unimodal	F1	Shifted and Rotated Bent Cigar	[-100,100]	100
	F2	Shifted and Rotated Sum of Different Power	[-100,100]	200
	F3	Shifted and Rotated Zakharov	[-100,100]	300
	F4	Shifted and Rotated Rosenbrock	[-100,100]	400
	F5	Shifted and Rotated Rastrigin	[-100,100]	500
Multimodal	F6	Shifted and Rotated Expanded Schafer	[-100,100]	600
	F7	Shifted and Rotated Lunacek Bi-kastrigin	[-100,100]	700
	F8	Shifted and Rotated Non-Continuous Rastrigin	[-100,100]	800
	F9	Shifted and Rotated Levy	[-100,100]	900
	F10	Shifted and Rotated Schwefel	[-100,100]	1000

Figure 15 shows the partial function convergence plots of the aforementioned algorithms on optimizing the CEE2017 test set for both single-modal and multimodal test functions, where the vertical axis and horizontal axis represent the optimal function values and iteration counts, respectively.

As shown in Figure 17, an improved SSA performs prominently in solving single-modal functions F3 and F4, slightly trailing ISSA and GWO algorithms. ISSA excels on single-modal functions with faster convergence speed and higher solution accuracy. In solving the multimodal function F5, an improved SSA demonstrates superior early optimization efficiency, finding a better solution space in just 170 iterations compared to other algorithms. In the mid-stage of the search, due to advantages in Cauchy mutation perturbation and stochastic direction learning mechanisms, an improved SSA exhibits advanced mobility, effectively preventing premature convergence and local optima traps. Therefore, around 200 iterations, the algorithm shows a turning point in optimization convergence, indicating that an improved SSA achieves high-precision global optimal solutions with enhanced exploration capability. However, an improved SSA performs poorly in solving the multimodal function F6, where ISPSO excels with faster convergence speed and higher solution accuracy. Conversely, an improved SSA performs superiorly in solving the multimodal function F4, achieving higher solution accuracy.

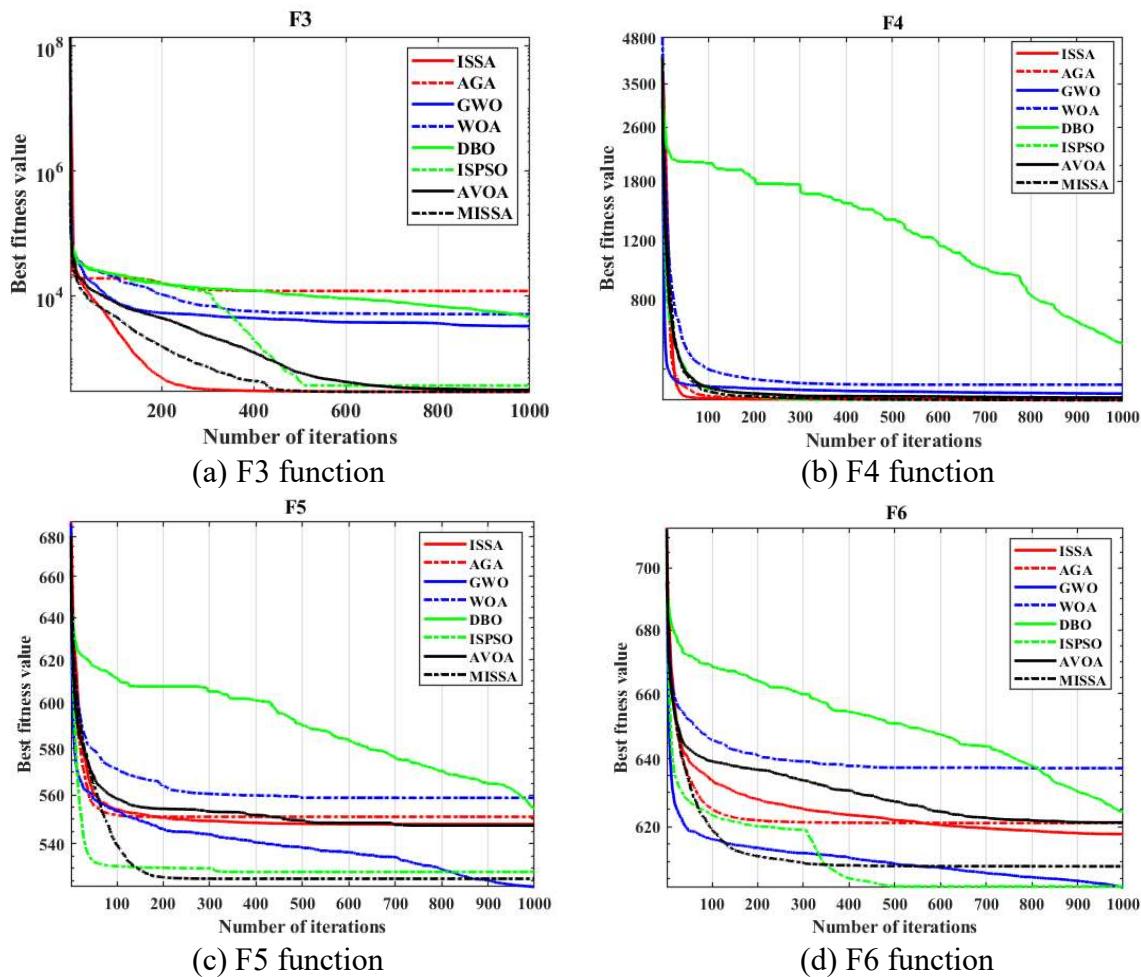


Figure 17. Variation results of best fitness for different algorithms.

Table 8 records the final computational results of optimization tests for six different algorithms, including optimal solutions, average solutions, and variance indicators. The size of the variance indicator reflects the stability of the algorithm in optimizing functions. In single-modal function performance tests, an improved SSA shows mixed results, achieving the best performance only on function F3. In contrast, ISSA, ISPSO, and WOA perform well on individual functions. In multimodal function tests, an improved SSA excels, performing best among the six test functions with higher stability, far exceeding other algorithms. For a more comprehensive assessment of these algorithms' overall performance, this paper conducts nonparametric tests on the results in Table 8, detailed in Figure 18.

Table 8. Computational results obtained by different algorithms on the CEC2017.

$F(x)$	Indicators	ISSA	AGA	WOA	ISPSO	AVOA	an improved SSA
F1	Best	1.32E+02	2.02E+02	5.33E+02	2.05E+02	5.35E+02	1.78E+02
	Mean	1.72E+02	3.22E+02	8.41E+02	3.43E+02	7.09E+02	1.92E+02
	Std	1.80E+03	2.21E+03	1.48E+03	6.56E+02	9.44E+02	2.03E+02
F2	Best	2.02E+02	2.04E+02	2.02E+02	2.02E+02	3.24E+02	2.09E+02
	Mean	2.03E+02	2.22E+02	2.01E+02	2.35E+02	5.55E+02	2.12E+02
	Std	2.25E+02	2.40E+02	2.32E+02	2.33E+02	4.23E+02	2.13E+02
F3	Best	3.05E+02	3.28E+02	3.75E+02	3.12E+02	3.22E+02	3.02E+02
	Mean	3.22E+02	1.75E+03	4.17E+04	3.43E+02	3.57E+02	3.13E+02
	Std	3.14E+01	2.75E+02	4.68E+03	2.24E+01	4.59E+01	2.14E+01
F4	Best	4.04E+02	4.02E-02	4.04E+02	4.01E+02	4.72E+02	4.00E+02
	Mean	4.26E+02	4.74E+02	4.51E+02	4.69E+02	4.66E+02	4.23E+02
	Std	5.23E+01	1.32E+02	3.62E+02	3.89E+02	5.21E+01	4.31E+02
F5	Best	5.29E+02	5.23E+02	5.24E+02	5.14E+02	6.25E+02	5.11E+02
	Mean	5.35E+02	5.34E+02	5.60E+02	5.41E+02	9.92E+02	5.38E+02
	Std	1.38E+01	5.89E+00	8.72E+00	1.23E+01	2.47E+01	1.09E+01
F6	Best	6.22E+02	6.11E+02	6.23E+02	6.31E+02	6.72E+02	6.07E+02
	Mean	6.71E+02	6.03E+02	6.42E+02	6.26E+02	8.21E+02	6.18E+02
	Std	7.59E+00	2.62E-02	2.33E+01	1.42E-01	1.46E+00	1.45E-01
F7	Best	7.27E+02	7.33E+02	7.78E+02	7.16E+02	7.91E+02	7.09E+02
	Mean	7.36E+02	7.25E+02	7.43E+02	7.14E+02	7.95E+02	7.17E+02
	Std	4.25E-01	3.54E-01	9.56E-01	2.33E+00	1.09E+00	2.63E-01
F8	Best	8.25E+02	8.19E+02	8.11E+02	8.11E+02	8.39E+02	8.05E+02
	Mean	8.32E+02	8.14E+02	8.13E+02	8.68E+02	8.32E+02	8.16E+02
	Std	3.43E+00	2.42E-01	4.92E-01	9.35E-01	1.26E+00	1.27E+00
F9	Best	9.02E+02	9.03E+02	9.03E+02	9.06E+02	1.72E+02	9.41E+02
	Mean	9.13E+02	9.32E+02	9.32E+02	1.19E+03	9.92E+02	1.17E+03
	Std	2.86E-03	4.56E+00	2.46E+00	2.48E+01	2.35E-01	4.25E+01
F10	Best	1.12E+03	1.41E+03	1.29E+03	1.53E+03	3.27E+03	1.07E+03
	Mean	1.87E+03	1.66E+03	1.51E+03	1.62E+03	4.08E+03	1.04E+03
	Std	2.52E+01	1.93E+00	2.59E-01	5.56E+00	2.88E+02	1.34E-01

Figure 18 displays the ranking results of all algorithms on the CEC2017 test functions. An improved SSA outperforms other algorithms significantly in both groups of CEC2017 functions. It achieves first place rankings in both single-modal and multimodal function categories. Therefore, it can be concluded that an improved SSA performs best on CEC2017, optimizing to better solution spaces compared to other algorithms, thus potentially replacing SSA for optimization tasks. The effectiveness and superiority of the proposed algorithm are validated accordingly.

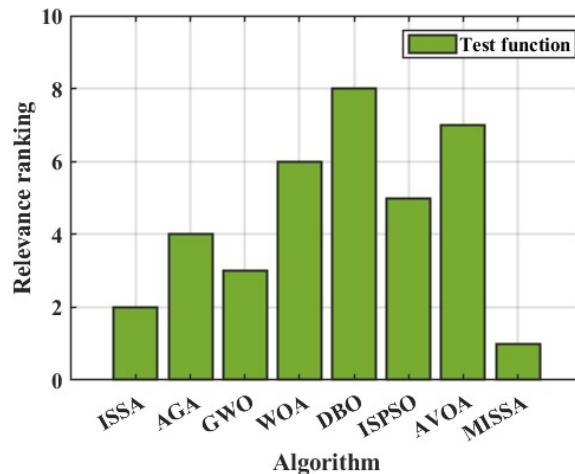


Figure 18. Nonparametric test results of CEC2017 test functions.

(3) Application analysis

The an improved SSA algorithm is applied to optimize the selection of hyperparameters for the CNN-Transformer-BiLSTM prediction model. Therefore, the performance of sewage flow prediction after optimizing hyperparameter selection reflects the applicability and superiority of the proposed algorithm. We apply different algorithms, including SSA and an improved SSA, to optimize hyperparameters of the CNN-Transformer-BiLSTM prediction model, and present their fitness value variations and evaluation metric results of the prediction model in Figure 17 and Table 8.

Figure 19 shows the fitness value variations of the SSA algorithm and an improved SSA algorithm at the same number of iterations. From the Figure 19, it can be observed that the fitness value decreases more rapidly with an improved SSA algorithm compared to the SSA algorithm, stabilizing earlier at the 5th iteration than at the 7th iteration for SSA, demonstrating better convergence. Additionally, the final fitness value optimized by an improved SSA algorithm is significantly lower than that of the SSA algorithm, indicating that an improved SSA can find the optimal solution faster during the optimization process, showcasing stronger global search capabilities and convergence efficiency.

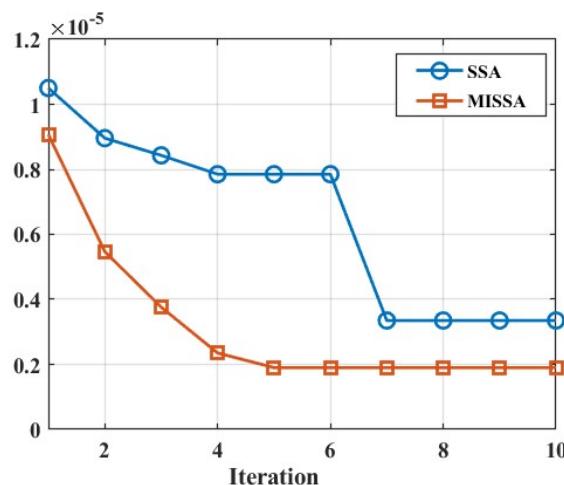


Figure 19. Variation of fitness values for SSA and an improved SSA optimization hyperparameters.

Comparing the evaluation metric results in Table 9, it can be seen that an improved SSA-CNN-Transformer-BiLSTM model achieves an R^2 score of 0.92603, an improvement over the CNN-Transformer-BiLSTM model's 0.85328 and the SSA-CNN-Transformer-BiLSTM model's 0.89545. This demonstrates that an improved SSA algorithm significantly enhances the predictive capability of the model when optimizing hyperparameter selection. The improved SSA-optimized model also performs well in terms of $RMSE$ and MAE , with values of 130.6226 and 94.0413, respectively, reducing by approximately 50 and 25 units compared to other models. This indicates that an improved SSA algorithm helps reduce prediction errors and improves the accuracy of sewage flow prediction. The improved SSA algorithm may be more effective in global search compared to the SSA algorithm, thereby avoiding local optima. This is particularly crucial for enhancing the model's generalization and stability, especially in complex time-series data prediction tasks.

Table 9. Results of indicator assessment for different forecasting models.

Model name	R^2	$RMSE$	MAE
CNN-Transformer-BiLSTM	0.85328	183.9647	145.3357
SSA-CNN-Transformer-BiLSTM	0.89545	155.2961	102.5557
an improved SSA-CNN-Transformer-BiLSTM	0.92603	130.6226	94.0413

4.5. Comparative experiments

To demonstrate the superiority of the constructed and improved SSA-CNN-Transformer-BiLSTM prediction model, this study also introduces four other combined prediction models for sewage flow prediction experiments. These prediction models include: KOA (kepler optimization algorithm)-CNN-BiGRU-SE (squeeze and excitation) model [46], CNN-LSTM-SE model [47], GWO-LSTM-SE model [48], and CNN-BiGRU-MH (multi-head) model [49]. The first 9544 samples of sewage flow data are used for model training, while the last 1024 data samples are used for testing the model's prediction outputs. The experiment is repeated five times, removing the best and worst prediction results, recording the average sewage flow predictions from the remaining three experiments, and using this data to plot the flow prediction curve of the models, as shown in Figure 20.

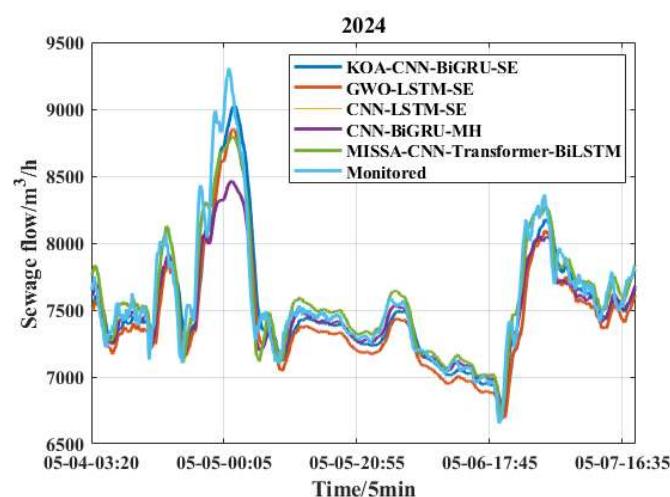


Figure 20. Variation curves of sewage flow predicted by different combination models.

Based on Figure 21, it can be observed that the predictions of an improved SSA-CNN-Transformer-BiLSTM model are closest to the actual flow values, indicating that this model performs best in flow prediction. The KOA-CNN-BiGRU-SE and GWO-LSTM-SE models follow closely in second and third positions, demonstrating that intelligent optimization algorithms improve the flow prediction capabilities of combined neural network models. Additionally, Figure 21 shows the relative errors of these prediction models on 40 sewage flow samples.

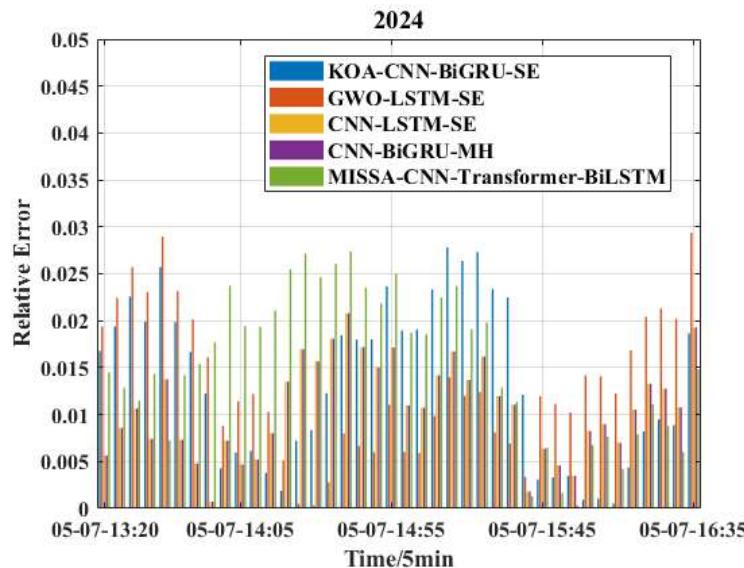


Figure 21. Histogram of relative errors in predicting sewage flows for different models.

According to Figure 21, the area of the green bars representing relative errors in the predictions of an improved SSA-CNN-Transformer-BiLSTM model is the smallest, followed by the KOA-CNN-BiGRU-SE model. This suggests that an improved SSA-CNN-Transformer-BiLSTM model has the best prediction performance, followed by the KOA-CNN-BiGRU-SE model, while the performance of the other models is relatively average.

From Table 10, it can be seen that the R^2 indicators of various models for predicting sewage flow output results, from smallest to largest, are as follows: CNN-LSTM-SE model, CNN-BiGRU-MH model, GWO-LSTM-SE model, PSO-GA-BP model, and an improved SSA-CNN-Transformer-BiLSTM model. Additionally, an improved SSA-CNN-Transformer-BiLSTM model achieves the minimum values for MAE , and $RMSE$ indicators. Therefore, an improved SSA-CNN-Transformer-BiLSTM model is closer to the real sewage flow data compared to the other four models, demonstrating the best prediction performance and validating the superiority of the proposed model.

Table 10. Results of indicator assessment for different forecasting models.

Indicator	an improved SSA-CNN-Transformer-BiLSTM	KOA-CNN-BiGRU-SE	CNN-LSTM-SE	GWO-LSTM-SE	CNN-BiGRU-MH
MAE	94.04131	108.52029	129.85206	116.33571	132.85206
$RMSE$	130.62268	138.25524	189.69534	143.96479	169.69535
R^2	0.92603	0.90859	0.84401	0.89328	0.86403

5. Conclusions

To address the complex problem of sewage flow prediction, this study proposes a combined prediction model based on ISSA, CNN, transformer, and BiLSTM for sewage flow prediction. Through this multimodel fusion approach, it is expected to overcome the limitations of individual models and achieve efficient and accurate prediction of sewage flow.

The proposed and improved SSA-CNN-Transformer-BiLSTM model is applied to a real sewage plant's sewage flow dataset, and experimental results demonstrate that the introduced Transformer mechanism effectively captures global dependencies, enhancing the handling of long-term sequential data. The proposed and improved SSA algorithm exhibits excellent global search capabilities and convergence efficiency, effectively optimizing the hyperparameter selection problem of the CNN-BiGRU-CBAM (convolutional block attention module), improving prediction capability, and reducing human interference. After introducing the improved SSA, CNN, and Transformer modules, the prediction model's R^2 increased by 0.18744, $RMSE$ decreased by 114.93, and MAE decreased by 86.67. The applicability and effectiveness of the model are validated. Compared with other combined prediction models, the proposed and improved SSA-CNN-Transformer-BiLSTM model achieves the highest prediction accuracy and higher interval coverage rate, demonstrating stronger model generalization capability and competitiveness. However, there are limitations to this model. For example, abnormal sewage flow due to external factors such as broken sewer pipes may not be predicted. That is what we will work on later.

Future work includes: (1) Further optimizing model performance: Although the proposed model performs well in the current study, there is space for further optimization. For example, exploring different network architectures or integrating more advanced technologies to enhance prediction accuracy and generalization capability. (2) Expanding application scenarios: Consider applying the proposed model to broader practical scenarios, such as other sewage treatment plants or different regions for sewage flow prediction. This would validate the model's applicability in diverse environments and further demonstrate its effectiveness and practicality.

Author contributions

Jiawen Ye: Conceptualization, Data curation, Investigation, Methodology, Writing-original draft, Visualization, Funding acquisition; Lei Dai: Formal analysis, Methodology, Writing-original draft, Validation, Writing-review & editing; Haiying Wang: Conceptualization, Methodology, Supervision, Funding acquisition, Writing review & editing. All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

The authors are much grateful to the anonymous referees for their constructive and substantive comments on our paper, which have considerably improved its presentation and quality.

This work is supported by 2024 University Students' Innovation and Entrepreneurship Training Project of China University of Geosciences, Beijing (Grant No. 202411415103). And it is also supported by 2024 Special Projects for Graduate Education and Teaching Reform from China University of Geosciences, Beijing (Grant No. JG2024021 and No. JG2024013), and 2024 Subject Development Research Fund Project of China University of Geosciences, Beijing (Grant No. 2024XK208).

Conflict of interest

The authors declare no conflict of interest.

References

- W. Chen, J. Lim, S. Miyata, Y. Akashi, Exploring the spatial distribution for efficient sewage heat utilization in urban areas using the urban sewage state prediction model, *Appl. Energ.*, **360** (2024), 122776. <https://doi.org/10.1016/J.APENERGY.2024.122776>
- Z. Jaffari, S. Na, A. Abbas, K. Y. Park, K. H. Cho, Digital imaging-in-flow (FlowCAM) and probabilistic machine learning to assess the sonolytic disinfection of cyanobacteria in sewage wastewater, *J. Hazard. Mater.*, **468** (2024), 133762. <https://doi.org/10.1016/J.JHAZMAT.2024.133762>
- A. Osmane, K. Zidan, R. Benaddi, S. Sbahi, N. Ouazzani, M. Belmouden, et al., Assessment of the effectiveness of a full-scale trickling filter for the treatment of municipal sewage in an arid environment: Multiple linear regression model prediction of fecal coliform removal, *J. Water Process Eng.*, **64** (2024), 105684. <https://doi.org/10.1016/j.jwpe.2024.105684>
- M. Ansari, F. Othman, A. El-Shafie, Optimized fuzzy inference system to enhance prediction accuracy for influent characteristics of a sewage treatment plant, *Sci. Total Environ.*, **722** (2020), 137878. <https://doi.org/10.1016/j.scitotenv.2020.137878>
- X. Wang, B. Zhao, X. Yang, Co-pyrolysis of microalgae and sewage sludge: Biocrude assessment and char yield prediction, *Energy Convers. Manage.*, **117** (2016), 326–334. <https://doi.org/10.1016/j.enconman.2016.03.013>
- V. Nourani, R. Zonouz, M. Dini, Estimation of prediction intervals for uncertainty assessment of artificial neural network based wastewater treatment plant effluent modeling, *J. Water Process Eng.*, **55** (2023), 104145. <https://doi.org/10.1016/j.jwpe.2023.104145>
- H. Mahanna, N. EL-Rahsidy, M. Kaloop, S. El-Sapakh, A. Alluqmani, R. Hassan, Prediction of wastewater treatment plant performance through machine learning techniques, *Desalin. Water Treat.*, **14** (2024), 100524. <https://doi.org/10.1016/j.dwt.2024.100524>
- J. Li, K. Sharma, Y. Liu, G. Jiang, Z. Yuan, Real-time prediction of rain-impacted sewage flow for on-line control of chemical dosing in sewers, *Water Res.*, **149** (2019), 311–321. <https://doi.org/10.1016/j.watres.2018.11.021>
- Y. Liu, X. Wu, W. Qi, Assessing the water quality in urban river considering the influence of rainstorm flood: A case study of Handan city, China, *Ecol. Indic.*, **160** (2024), 111941. <https://doi.org/10.1016/j.ecolind.2024.111941>
- E. Ekinci, B. Özbay, S. Omurca, F. Sayın, İ. Özbay, Application of machine learning algorithms and feature selection methods for better prediction of sludge production in a real advanced biological wastewater treatment plant, *J. Environ. Manag.*, **348** (2023), 119448. <https://doi.org/10.1016/j.jenvman.2023.119448>
- Z. Gao, J. Chen, G. Wang, S. Ren, L. Fang, Y. Aa, et al., A novel multivariate time series prediction of crucial water quality parameters with Long Short-Term Memory (LSTM) networks, *J. Contam. Hydrol.*, **259** (2023), 104262. <https://doi.org/10.1016/j.jconhyd.2023.104262>
- M. Yaqub, H. Asif, S. Kim, W. Lee, Modeling of a full-scale sewage treatment plant to predict the nutrient removal efficiency using a long short-term memory (LSTM) neural network, *J. Water Process Eng.*, **37** (2020), 101388. <https://doi.org/10.1016/j.jwpe.2020.101388>

13. N. Farhi, E. Kohen, H. Mamane, Y. Shavitt, Prediction of wastewater treatment quality using LSTM neural network, *Environ. Technol. Inno.*, **23** (2021), 101632. <https://doi.org/10.1016/j.eti.2021.101632>
14. W. Alfwzan, M. Selim, A. Almalki, I. S. Alharbi, Water quality assessment using Bi-LSTM and computational fluid dynamics (CFD) techniques, *Alex. Eng. J.*, **97** (2024), 346–359. <https://doi.org/10.1016/j.aej.2024.04.030>
15. W. Zhang, J. Zhao, P. Quan, J. Wang, X. Meng, Q. Li, Prediction of influent wastewater quality based on wavelet transform and residual LSTM, *Appl. Soft Comput.*, **148** (2023), 110858. <https://doi.org/10.1016/j.asoc.2023.110858>
16. Y. Zhang, C. Li, Y. Jiang, L. Sun, R. Zhao, K. Yan, et al., Accurate prediction of water quality in urban drainage network with integrated EMD-LSTM model, *J. Clean. Prod.*, **354** (2022), 131724. <https://doi.org/10.1016/j.jclepro.2022.131724>
17. L. Zheng, H. Wang, C. Liu, S. Zhang, A. Ding, E. Xie, et al., Prediction of harmful algal blooms in large water bodies using the combined EFDC and LSTM models, *J. Environ. Manag.*, **295** (2021), 113060. <https://doi.org/10.1016/j.jenvman.2021.113060>
18. L. Zhang, C. Wang, W. Hu, X. Wang, H. Wang, X. Sun, et al., Dynamic real-time forecasting technique for reclaimed water volumes in urban river environmental management, *Environ. Res.*, **248** (2024), 118267. <https://doi.org/10.1016/j.envres.2024.118267>
19. S. Huan, A novel interval decomposition correlation particle swarm optimization-extreme learning machine model for short-term and long-term water quality prediction, *J. Hydrol.*, **625** (2023), 130034. <https://doi.org/10.1016/j.jhydrol.2023.130034>
20. H. Darabi, A. Haghghi, O. Rahmati, A. Shahrood, S. Rouzbeh, B. Pradhan, et al., A hybridized model based on neural network and swarm intelligence-grey wolf algorithm for spatial prediction of urban flood-inundation, *J. Hydrol.*, **603** (2021), 126854. <https://doi.org/10.1016/j.jhydrol.2021.126854>
21. Z. Wang, H. Dai, B. Chen, S. Cheng, Y. Sun, J. Zhao, et al., Effluent quality prediction of the sewage treatment based on a hybrid neural network model: Comparison and application, *J. Environ. Manag.*, **351** (2024), 119900. <https://doi.org/10.1016/j.jenvman.2023.119900>
22. F. Farzin, S. Moghaddam, M. Ehteshami, Auto-tuning data-driven model for biogas yield prediction from anaerobic digestion of sewage sludge at the south-tehran wastewater treatment plant: Feature selection and hyperparameter population-based optimization, *Renew. Energy*, **227** (2024), 120554. <https://doi.org/10.1016/j.renene.2024.120554>
23. G. Ye, J. Wan, Z. Deng, Y. Wang, J. Chen, B. Zhu, et al., Prediction of effluent total nitrogen and energy consumption in wastewater treatment plants: Bayesian optimization machine learning methods, *Bioresource Technol.*, **395** (2024), 130361. <https://doi.org/10.1016/j.biortech.2024.130361>
24. J. Piri, B. Pirzadeh, B. Keshtegar, M. Givehchi, Reliability analysis of pumping station for sewage network using hybrid neural networks-genetic algorithm and method of moment, *Process Saf. Environ.*, **145** (2021), 39–51. <https://doi.org/10.1016/j.psep.2020.07.045>
25. M. Salamattalab, M. Zonoozi, M. Molavi-Arabshahi, Innovative approach for predicting biogas production from large-scale anaerobic digester using long-short term memory (LSTM) coupled with genetic algorithm (GA), *Waste Manag.*, **175** (2024), 30–41. <https://doi.org/10.1016/j.wasman.2023.12.046>
26. A. Mohammed, K. Hassan, M. Abdel-Aal, Moving average smoothing for gregory-newton interpolation: A novel approach for short-term demand forecasting, *IFAC-PapersOnLine*, **55** (2022), 749–754. <https://doi.org/10.1016/j.ifacol.2022.09.499>
27. P. Mei, M. Li, Q. Zhang, G. Li, L. Song, Prediction model of drinking water source quality with potential industrial-agricultural pollution based on CNN-GRU-Attention, *J. Hydrol.*, **610** (2022), 127934. <https://doi.org/10.1016/j.jhydrol.2022.127934>

28. A. L. de Rojas, M. A. Jaramillo-Morán, J. E. Sandubete, EMDFormer model for time series forecasting, *AIMS Math.*, **9** (2024), 9419–9434. <https://doi.org/10.3934/math.2024459>
29. H. Jin, Y. Liang, H. Lu, S. Zhang, Y. Gao, Y. Zhao, et al., An intelligent framework for spatiotemporal simulation of flooding considering urban underlying surface characteristics, *Int. J. Appl. Earth Obs.*, **130** (2024), 103908. <https://doi.org/10.1016/j.jag.2024.103908>
30. B. Qu, E. Jiang, J. Li, Y. Liu, C. Liu, Coupling coordination relationship of water resources, eco-environment and socio-economy in the water-receiving area of the Lower Yellow River, *Ecol. Indic.*, **160** (2024), 111766. <https://doi.org/10.1016/j.ecolind.2024.111766>
31. J. Cai, B. Sun, H. Wang, Y. Zheng, S. Zhou, H. Li, et al., Application of the improved dung beetle optimizer, multi-head attention and hybrid deep learning algorithms to groundwater depth prediction in the Ningxia area, China, *Atmos. Ocean. Sci. Lett.*, 2024, 100497. <https://doi.org/10.1016/j.aosl.2024.100497>
32. M. Wang, G. Zhao, S. Wang, Hybrid random forest models optimized by Sparrow search algorithm (SSA) and Harris hawk optimization algorithm (HHO) for slope stability prediction, *Transp. Geotech.*, **48** (2024), 101305. <https://doi.org/10.1016/j.trgeo.2024.101305>
33. C. Zhang, S. Ding, A stochastic configuration network based on chaotic sparrow search algorithm, *Knowl.-Based Syst.*, **220** (2024), 106924. <https://doi.org/10.1016/j.knosys.2021.106924>
34. X. Shao, J. Yu, Z. Li, X. Yang, B. Sundén, Energy-saving optimization of the parallel chillers system based on a multi-strategy improved sparrow search algorithm, *Heliyon*, **9** (2023), e21012. <https://doi.org/10.1016/j.heliyon.2023.e21012>
35. X. Long, W. Cai, L. Yang, H. Huang, Improved particle swarm optimization with reverse learning and neighbor adjustment for space surveillance network task scheduling, *Swarm Evol. Comput.*, **85** (2024), 101482. <https://doi.org/10.1016/j.swevo.2024.101482>
36. S. Zhao, Y. Duan, N. Roy, B. Zhang, A deep learning methodology based on adaptive multiscale CNN and enhanced highway LSTM for industrial process fault diagnosis, *Reliab. Eng. Syst. Safe.*, **249** (2024), 110208. <https://doi.org/10.1016/j.ress.2024.110208>
37. K. Wang, X. Fan, X. Yang, Z. Zhou, An AQI decomposition ensemble model based on SSA-LSTM using improved AMSSA-VMD decomposition reconstruction technique, *Environ. Res.*, **232** (2023), 116365. <https://doi.org/10.1016/j.envres.2023.116365>
38. Y. Leng, H. Zhang, X. Li, A novel evaluation method for renewable energy development based on improved sparrow search algorithm and projection pursuit model, *Expert Syst. Appl.*, **244** (2024), 122991. <https://doi.org/10.1016/j.eswa.2023.122991>
39. Z. Zhang, X. Cheng, Z. Xing, Z. Wang, Y. Qin, Optimal sizing of battery-supercapacitor energy storage systems for trams using improved PSO algorithm, *J. Energy Storage*, **73** (2023), 108962. <https://doi.org/10.1016/j.est.2023.108962>
40. J. Li, R. Liu, R. Wang, Handling dynamic capacitated vehicle routing problems based on adaptive genetic algorithm with elastic strategy, *Swarm Evol. Comput.*, **86** (2024), 101529. <https://doi.org/10.1016/j.swevo.2024.101529>
41. X. Zhang, J. Xia, Z. Chen, J. Zhu, H. Wang, A nutrient optimization method for hydroponic lettuce based on multi-strategy improved grey wolf optimizer algorithm, *Comput. Electron. Agr.*, **224** (2024), 109167. <https://doi.org/10.1016/j.compag.2024.109167>
42. J. Sahayaraj, K. Gunasekaran, S. Verma, M. Dhurgadevi, Energy efficient clustering and sink mobility protocol using improved dingo and boosted beluga whale optimization algorithm for extending network lifetime in WSNs, *Sustain. Comput.-Infor.*, **43** (2024), 101008. <https://doi.org/10.1016/j.suscom.2024.101008>

43. F. Zhu, G. Li, H. Tang, Y. Li, X. Lv, X. Wang, Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems, *Expert Syst. Appl.*, **236** (2024), 121219. <https://doi.org/10.1016/j.eswa.2023.121219>

44. L. Yin, W. Ding, Deep neural network accelerated-group african vulture optimization algorithm for unit commitment considering uncertain wind power, *Appl. Soft Comput.*, **162** (2024), 111845. <https://doi.org/10.1016/j.asoc.2024.111845>

45. M. Abdel-Basset, R. Mohamed, M. Abouhawwash, Crested porcupine optimizer: A new nature-inspired metaheuristic, *Knowl.-Based Syst.*, **284** (2024), 111257. <https://doi.org/10.1016/j.knosys.2023.111257>

46. U. Khan, N. Khan, M. Zafar, Resource efficient PV power forecasting: Transductive transfer learning based hybrid deep learning model for smart grid in Industry 5.0, *Energy Convers. Man.-X*, **20** (2024), 100486. <https://doi.org/10.1016/j.ecmx.2023.100486>

47. X. Zhou, B. Sheil, S. Suryasentana, P. Shi, Multi-fidelity fusion for soil classification via LSTM and multi-head self-attention CNN model, *Adv. Eng. Inform.*, **62** (2024), 102655. <https://doi.org/10.1016/j.aei.2024.102655>

48. M. Javanmard, S. Ghaderi, A hybrid model with applying machine learning algorithms and optimization model to forecast greenhouse gas emissions with energy market data, *Sustain. Cities Soc.*, **82** (2022), 103886. <https://doi.org/10.1016/j.scs.2022.103886>

49. S. Tariq, J. Loy-Benitez, K. Nam, S. Kim, M. Kim, C. Yoo, Deep-AI soft sensor for sustainable health risk monitoring and control of fine particulate matter at sesnsor devoid underground spaces: A zero-shot transfer learning approach, *Tunn. Undergr. Sp. Tech.*, **131** (2023), 104843. <https://doi.org/10.1016/j.tust.2022.104843>

50. Z. Wang, N. Xu, X. Bao, J. Wu, X. Cui, Spatio-temporal deep learning model for accurate streamflow prediction with multi-source data fusion, *Environ. Modell. Softw.*, **178** (2024), 106091. <https://doi.org/10.1016/j.envsoft.2024.106091>

51. G. Dai, Z. Tian, J. Fan, C. K. Sunil, C. Dewi, DFN-PSAN: Multi-level deep information feature fusion extraction network for interpretable plant disease classification, *Comput. Electron. Agr.*, **216** (2024), 108481. <https://doi.org/10.1016/j.compag.2023.108481>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)