International Journal of Innovative Computing, Information and Control
Volume **21**, Number **5**, October **2025**

ICIC International ©2025 ISSN 1349-4198
pp. **1187–1208**

MULTI-FEATURE WATER CONSUMPTION TIME SERIES PREDICTION METHOD BASED ON OPERATING CONDITION MIGRATION

Fenglong Kan[1,*], Weiyang Li[1] and Guowei Yuan[2]

[1]School of Electrical and Control Engineering
Shenyang Jianzhu University
No. 25, Hunnan Middle Road, Hunnan District, Shenyang 110168, P. R. China
liweiyang@sjzu.edu.cn
*Corresponding author: kanfenglong@sjzu.edu.cn

[2]School of Mechanical Engineering
Liaoning Mechanical & Electrical College of Technology
No. 30, Yanghe Street, Zhenxing District, Dandong 118009, P. R. China
yuanguowei@lnmec.net.cn

Received December 2024; revised April 2025

Abstract. *The prediction of water demand plays a crucial role in optimizing the scheduling decisions of water supply networks, facilitating a balance between supply and demand, and reducing the operating costs of water plants. This paper addresses the challenges associated with predicting water consumption, which is influenced by a large number of correlated features, non-stationary variations in those features, and fluctuations in operational conditions throughout the water consumption process. Aiming at these challenges, the paper proposes a multi-feature water consumption time series prediction method based on operational condition migration. First, to address the issue of the large number of correlated features with inconsistent importance, factor analysis is used for dimensionality reduction on multi-dimensional water supply data, followed by K-means++ clustering to obtain visualized operating condition classifications and construct an intuitive and simplified low-dimensional dataset. Second, a multi-head self-attention mechanism is integrated into the Long Short-Term Memory (LSTM) networks to build prediction sub-models corresponding to each operating condition cluster. These sub-models can increase the weight of data points during prediction and reflect the effects of feature interactions. Finally, this paper proposes an operating condition migration multi-source path optimization algorithm, which combines both time-domain and space-domain information to obtain the water demand prediction value at the current moment and the globally optimal operating condition migration path. This paper conducts extensive experimental validation of the algorithm using two publicly available datasets and one self-constructed dataset. The experimental results show that the proposed method outperforms Transformer and LSTM-based methods in terms of prediction error and graphical fitting accuracy, particularly under conditions with a high number of missing values and feature absence.*

Keywords: Water consumption prediction, Factor analysis, LSTM, Multi-head self-attention mechanism, Operational condition migration

1. **Introduction.** Water consumption prediction refers to the process of forecasting and estimating the water supply, demand, or hydrological conditions for a future period based on certain models, algorithms, or methods [1]. Water consumption prediction helps water resource managers better plan and manage the supply and distribution of water resources.

DOI: 10.24507/ijicic.21.05.1187

By accurately predicting water consumption, the utilization of water resources can be optimized, ensuring the stability and sustainability of water supply [2]. Time series prediction is important for understanding future temporal information and is widely applied in fields such as traffic flow forecasting [3], economic analysis [4], industrial production regulation [5], and biomedicine [6]. Therefore, performing time series prediction for water consumption has significant practical importance.

In the field of time series prediction, both domestic and international researchers have conducted extensive studies. There are numerous methods for predicting time series data, which can be broadly categorized into two major types: traditional prediction methods and machine learning-based prediction methods. Typical traditional methods include Autoregressive (AR) [7] models, Moving Average (MA) [8] models, Autoregressive Moving Average (ARMA) [9] models, and Autoregressive Integrated Moving Average (ARIMA) [10] models. These traditional methods have the limitation of only being able to handle relatively smooth data. However, in real scenarios, time series data is often non-stationary, which means traditional methods typically suffer from low prediction accuracy. To address the limitations of traditional methods, machine learning has been widely applied to the prediction of non-stationary time series, such as Support Vector Machines (SVM) [11], clustering [12], and neural networks [13]. In recent years, the rise of deep learning has garnered significant attention, with Long Short-Term Memory (LSTM) networks being widely applied in the field of time series forecasting. Siłka et al. [14] proposed a time series forecasting method based on the Recurrent Neural Network (RNN) model, but due to issues such as vanishing and exploding gradients, the model faces difficulty in converging during training. The LSTM network, proposed by Schmidhuber and Hochreiter [15], solves the RNN convergence problem by adding a forget gate. LSTM networks are convenient for time series modeling and possess long-term memory capabilities, which has led to their widespread application in forecasting [16-18]. However, LSTM networks struggle to focus on different variables with varying time series lengths and cannot guarantee robust prediction performance during forecasting [19]. To address this issue, Cho et al. [20] proposed a water level prediction model using LSTM and Gated Recurrent Units (GRU). Through experiments with various model structures and different input data formats, the appropriate number of hidden layers for each case was selected, resulting in a water level prediction model that improves stability and generalization performance. Ghosh [21] utilized the self-attention mechanism of the Transformer model to focus on the most crucial features of the current task, enabling efficient and precise predictions. However, when the predicted time series exhibit local fluctuations and global periodic trends, the Transformer model is less effective in capturing the detailed characteristics of the sequence. This limitation becomes more pronounced in the prediction of water supply time series with short time intervals. Wen and Li [22] proposed an LSTM-attention-LSTM model that incorporates an attention mechanism between the encoder and decoder. This model demonstrated superior prediction performance compared to algorithms such as encoder-decoder LSTM, stacked LSTM, and BiLSTM when dealing with complex, large-scale, and noisy data. By integrating the attention mechanism into the LSTM, the decoder can focus on more important data points at the current time step while retaining the advantage of effectively handling long time series. Niknam et al. [23] investigated the impact of climate features on water consumption using the Pearson correlation coefficient, discarding factors with low correlation and inputting variables with higher correlation into the MV-LSTM prediction model. This approach enhanced the model's prediction speed and reduced training time under the influence of multiple environmental factors.

However, there are still the following issues in the research on the stable supply of hydraulic systems [24] and water resource scheduling and allocation [25].

1) Insufficient extraction of current operational condition features, with a limited consideration of operating conditions. In reality, water consumption falls into two scenarios: small variations within the same operational condition range and directional changes that cross over different operational condition ranges. Complete multi-dimensional data is difficult to obtain in actual production, and there are varying degrees of correlation between the different dimensions of data.

2) Short-term water consumption is influenced by multiple factors, such as the daily temperature, rainfall, and the periodicity of water usage during morning and evening. Each factor impacts the prediction with different time steps, reflected in the characteristic changes at time nodes in the dataset and the varying operating conditions during the water supply process. When predicting the water discharge at a specific time point, multiple factors are combined, and the contribution of each factor is not fixed.

3) Existing prediction methods have reduced the prediction error to some extent, but as the time step increases, the prediction accuracy decreases rapidly. When there are many missing values, the model training effect is poor due to the imputed data, and the overfitting phenomenon of the prediction output becomes more apparent as the time step grows.

To address the shortcomings of the above methods, this paper proposes a multi-feature water demand time series forecasting method based on operating condition migration. Its main contributions are as follows.

1) The proposed model applies factor analysis to reducing the dimensionality of multi-dimensional water supply data, addressing the issue of numerous related features with complex correlations [26]. Subsequently, K-means++ clustering is used to partition the operating conditions, generating clusters of operating conditions from the dataset. The method remains adaptable even when the dataset is missing several dimensions or when the dataset's dimensions are more complex than initially anticipated.

2) The proposed model incorporates the Multi-Head Self-Attention (MSA) mechanism between the encoding and decoding layers of the LSTM to construct the LSTM-MSA prediction submodel. When the attention mechanism allocates attention to the input time steps, it references the dataset of the same operating condition, assigning weights to different time steps based on the corresponding operating conditions.

3) The proposed model constructs a condition migration network and introduces a multi-source path optimization algorithm to solve for the optimal operating condition submodel's water consumption prediction at the current time step, based on the actual and predicted water consumption values. During the training process of the condition migration network, the impact of missing values on the model's prediction accuracy is minimized.

4) Extensive experiments conducted on two public datasets and one self-constructed dataset consistently show that the model proposed in this paper achieves state-of-the-art performances under conditions of data missing, insufficient feature, as well as for both short-term and long-term prediction tasks.

The remainder of this paper is organized as follows. Section 2 introduces the contents of three datasets and a method for preprocessing the collected data. Section 3 describes the construction process and operational sequence of the algorithm model proposed in this paper. Section 4 evaluates the prediction performance of recent models and the proposed model based on experiments with three datasets, as well as their applicability under different time step conditions. Section 5 summarizes the paper and presents future research directions.

2. **Multivariate Time Series Data Preprocessing.** In this section, the relationship between water consumption, temperature, and precipitation is first introduced, along with its connection to some water supply-related data. Then, the process of dimensionality reduction of high-dimensional datasets using factor analysis is described. Finally, clustering of the resulting low-dimensional dataset is performed to categorize operating conditions as part of the data preprocessing.

Three multivariate time series water consumption datasets are used: Public Dataset A (Samuel Street – South 9th Street, Samuel City) [27], Public Dataset B (Brooklyn District, New York City) [28], and Self-constructed Dataset C (a water utility company in Shenyang). 1) Dataset A comes from water consumption data collected by the Federal Housing Administration from 2013 to February 2023 in the Samuel City neighborhood, with a collection interval of 60 minutes, meaning data is collected once every hour. 2) Dataset B comes from water consumption and operational cost data for Brooklyn District, New York City, published by the New York City Housing Authority, covering the period from 2013 to 2020. The data collection interval is 15 minutes, meaning data is collected four times per hour. 3) Dataset C comes from the internal water plant private dataset of a water utility company in Shenyang, covering the period from 2023 to April 2024. The actual data detected by various onsite sensors is obtained through the PLC controller and SCADA system. The data collection interval is 5 minutes, meaning data is collected twelve times per hour.

In the experiment, the data from January 2021 to December 2022 of Dataset A is selected as the training set, and the data from January to February 2023 is used as the test set. The data missing rate is approximately 7.5%. For Dataset B, the training set covers January 2020 to November 2020, and the test set covers December 2020, with a missing data rate of approximately 2%. For Dataset C, the training set covers June 2023 to November 2023, and the test set covers December 2024, with a missing data rate of approximately 1.5%. Missing values in all three datasets are filled using the mean of adjacent data points. Dataset B is missing the feature of instantaneous flow for groundwater imports. After conducting the KMO test and Bartlett's sphericity test, Dataset B still meets the prerequisites for Principal Component Analysis (PCA). For each test set, the same time-step random slicing operation is performed internally to generate 10 random test subsets.

2.1. **Data correlation analysis.** Water consumption can be reflected from historical data related to the water supply chain, and is associated with the temperature of the day and the cumulative precipitation over the past 36 hours. Data related to the water supply chain typically includes various dimensions such as the water outlet pressure, pump group operating current, clear water tank level, groundwater valve opening, and NaClO hourly dosage. First, calculate the correlation between each feature and the correlation between features and the actual water output. The closer the correlation coefficient is to 1, the stronger the correlation between the two physical quantities. Referring to research articles on water consumption prediction and considering the ease of obtaining features from the actual water consumption dataset, this study selects eight key features with high correlation to water output as model inputs: water outlet pressure, instantaneous flow rate of the groundwater inlet, surface water valve opening, clear water tank level, pump group operating current, water output, daily temperature, and cumulative precipitation over the past 36 hours. The range of water outlet pressure can roughly indicate the water consumption under three different operating conditions. Including this feature ensures reliable data support and validation when clustering operating conditions in subsequent steps. The instantaneous flow rate of the groundwater inlet and the surface water valve

opening show a trend of leading changes in the data, while the clear water tank level exhibits obvious lag. These insights are beneficial for assigning attention weights to the operating condition sub-models in later algorithms and for finding the minimal migration cost path. The operating current of the pump group is a commonly recorded data point in most water plants when monitoring the water quantity process. This feature has significant reference value for predicting the magnitude and direction of water consumption changes. A segment of the 72-hour variations in water outlet pressure, pump group operating current, instantaneous flow rate of the groundwater inlet, and clear water tank level is shown in Figure 1. In the collected dataset, features such as NaClO tank level, hourly dosage, surface water pressure, and turbidity have a relatively low correlation with water consumption. Including these features would increase the computational load, slow down the model's running speed, and yield little improvement in performance. Moreover, there is a high probability that these features will be discarded in subsequent factor analysis validation steps due to their low relevance.

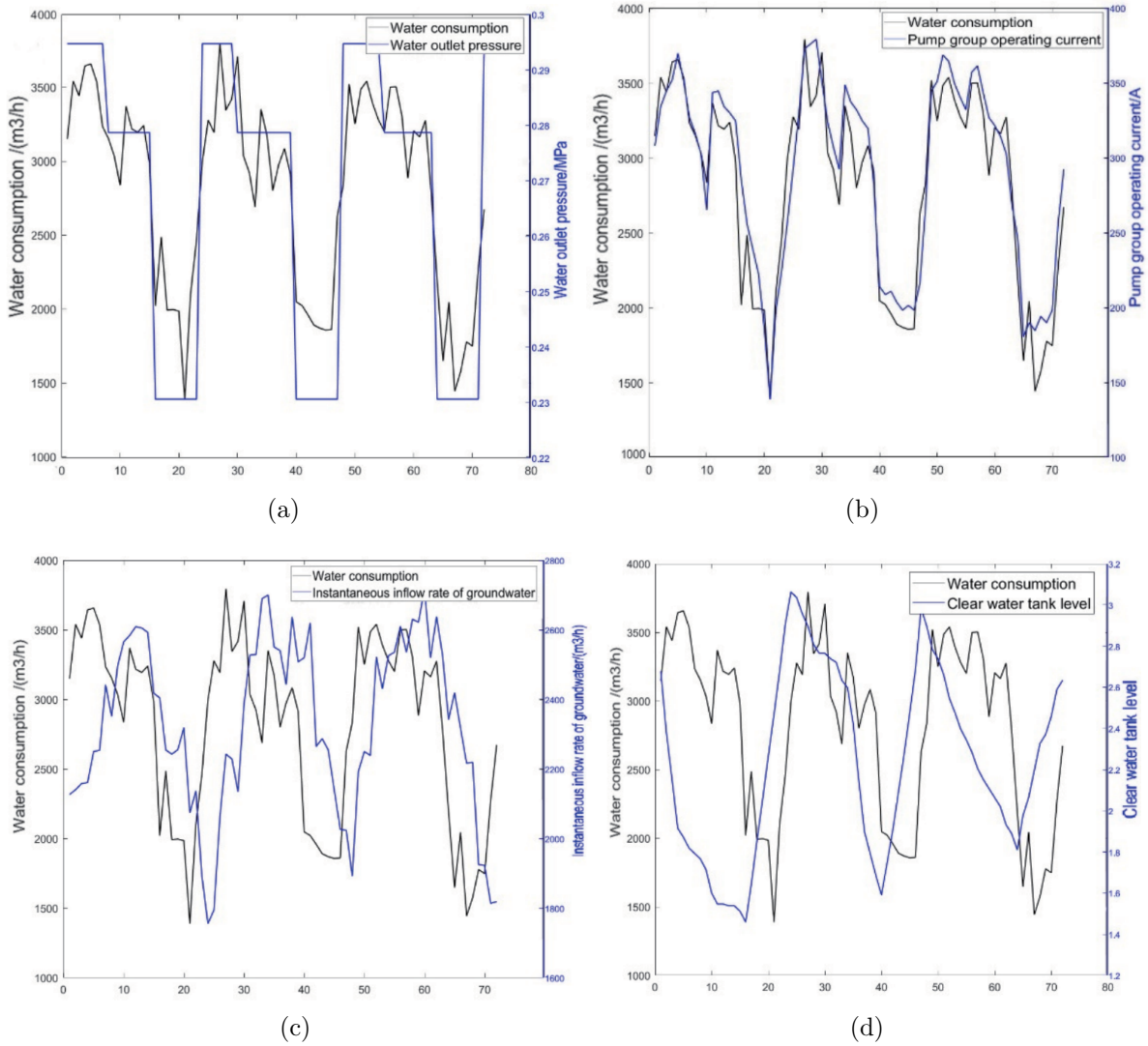


FIGURE 1. The variation of outlet pressure (a), pump group operating current (b), instantaneous inflow rate of groundwater (c), clear water tank level (d), and water consumption

2.2. **Factor analysis.** Factor Analysis (FA) is a multivariate statistical analysis method based on the concept of dimensionality reduction, aiming to reveal the underlying structure or relationships among observed variables. Factor analysis can be introduced to extract the effective features of water consumption, obtaining a set of common factors that are far fewer than the number of original features, which will then be used as input data for K-means++ clustering. A large number of variables are linearly represented by several independent common factors through a correlation coefficient matrix, identifying a small number of variables that reflect the relationships among the many variables.

2.3. **Water consumption condition clustering.** Dividing the conditions helps avoid the impact of the actual water consumption's imbalanced characteristics on the model and is a crucial step for allocating attention weights between the encoding and decoding layers of the subsequent LSTM model, based on different conditions. In this paper, the input process variable dataset is first reduced in dimensionality through factor analysis and then clustered using K-means++. The dataset is divided into clusters corresponding to different condition types. When allocating attention weights to the prediction submodels corresponding to these condition clusters, more weight will be assigned to data points within the same cluster.

The commonly used clustering evaluation metrics, the Silhouette Coefficient (SC) and the Adjusted Rand Index (ARI), are employed to determine the optimal number of clusters (K) in condition clustering. The dataset's clustering performance was evaluated using SC and ARI, and it was found that the optimal clustering occurs when $K = 3$, i.e., when the data is divided into three clusters. This result is consistent with our observations. Figure 2 shows the clustering results for dataset C from June 1 to June 7, 2023.

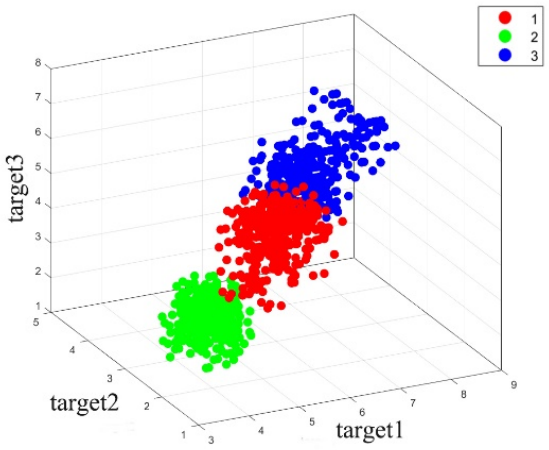


FIGURE 2. K-means++ clustering diagram after factor analysis

The factor analysis method used in this paper is applicable both when a key feature is missing in the dataset and when the collected features are abundant. It demonstrates better adaptability and selection capability in data preprocessing. When a feature is missing in the collected dataset, compute the correlation between the remaining features and the missing feature to identify the most correlated feature and reasonably replace the missing one. When the number of features collected in the dataset is richer than expected, correlation calculations are used to increase the number of features with a high correlation to water consumption and remove redundant features with low correlation.

3. **Algorithm Framework.** This paper proposes using the LSTM-MSA operational condition network model to predict the water consumption time series at each time step, as shown in Figure 3. The model structure consists of a data preprocessing module, input layer, LSTM encoding layer, individual attention layers, LSTM decoding layer, output layer, and fully connected layer. The input layer sends the data to the factor analysis clustering module and the encoding layer. The results of the factor analysis clustering module determine the size of the attention parameters for different sub-models. The encoding layer encodes the input sequence $X(X_1, X_2, X_3, \ldots, X_t)$, and then the attention layer calculates the attention weights, which are followed by the decoding layer that decodes the data. Finally, the fully connected layer connects the predicted sequence data from the output layer based on the multi-source path optimization algorithm to obtain $Y(Y_{t1}, Y_{t2}, Y_{t3}, \ldots, Y_{tm})$.

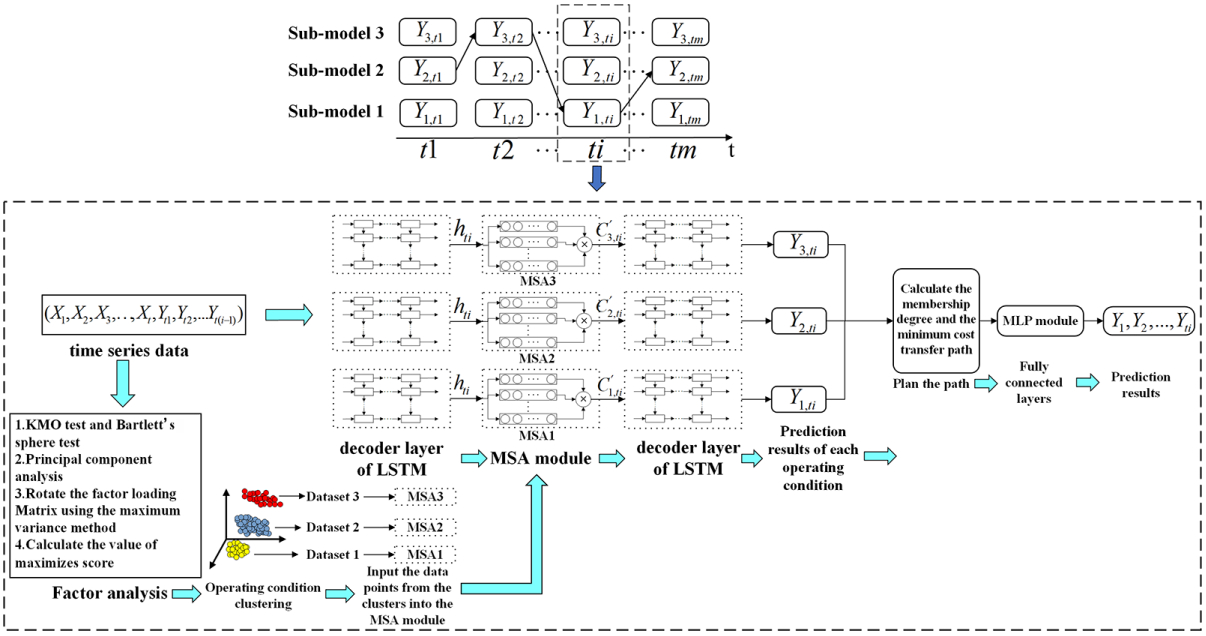


FIGURE 3. Overall prediction flowchart

The algorithm can retain the impact of historical data on current predictions through LSTM. At the same time, by combining the multi-head self-attention mechanism, it can allocate self-attention weights during the encoding-decoding process, incorporating features from different dimensions to obtain predictions for different operating conditions at the same time step. The use of the minimal cost migration path connection in the operational condition network can improve the curve fitting accuracy, closely matching the actual water consumption, thereby achieving more precise prediction results.

3.1. **LSTM.** Long Short-Term Memory (LSTM) introduces memory units based on Recurrent Neural Networks (RNN). A single LSTM neuron includes a forget gate, input gate, and output gate, as shown in Figure 4. LSTM retains the temporal features of sequential data through these three gate structures.

LSTM, due to its unique gating mechanism, is very effective at handling issues of gradient explosion or gradient vanishing, and it does not require fixed-length input sequences, making it highly suitable for handling situations where the time series length in water consumption prediction is variable.

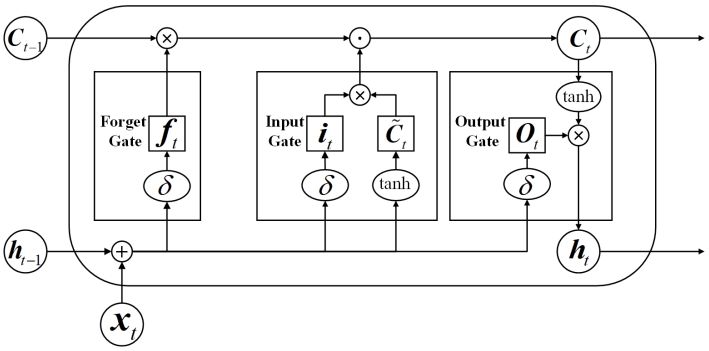


FIGURE 4. Structure of the LSTM unit

3.2. **Multi-head self-attention mechanism.** In the LSTM model, there is only a single link between the encoder and decoder models, and this link is a fixed-length vector. If the input sequence is long, earlier parts of the sequence may be forgotten as later parts are processed. Additionally, the fixed-length vector cannot fully represent the entire sequence. To address this issue, as shown in Figure 3, a Multi-Head Self-Attention (MSA) module is added between the LSTM encoder layer and decoder layer.

After incorporating the multi-head self-attention mechanism into the proposed model, multiple vectors collaboratively work to retain the crucial components of temporal features that are important for water consumption prediction. This approach helps prevent the loss of critical information during the training process, which is a common issue in LSTM models as the length of the time series increases. The multi-head self-attention module allocates attention weights based on the corresponding clusters of input data, which enhances the predictive performance of sub-models under different operating conditions and further improves the overall model's prediction accuracy.

3.3. **Optimal condition-based sub-model water consumption prediction.**

3.3.1. *LSTM-attention sub-model design.* Considering the numerous successful precedents of LSTM-based models in time series prediction tasks, and the widespread application of attention mechanisms in natural language processing and computer vision, this paper proposes a predictive sub-model that combines the LSTM model with the attention mechanism. It leverages the advantages of LSTM in handling time series data and the ability of the attention mechanism to allocate attention weights, to predict the input time series and fully utilize the strengths of each module.

The input layer encodes the input $X(X_1, X_2, X_3, \ldots, X_t)$, and the encoding layer is derived from LSTM. The calculation for the encoding layer is as follows:

$$h_i = f(X_i, h_{i-1}) \quad 1 \leq i \leq t \tag{1}$$

where h_i is the hidden state at time step i calculated by the encoding layer, h_{i-1} is the hidden state at time step $i-1$, t is the length of the input sequence, and f is the computation function for the input gate, forget gate, and output gate of the LSTM model.

After the encoding layer, batch normalization is applied, and the output is fed into the attention layer. The attention mechanism calculates the attention weight α_{ij} for the sequence at time step i of sequence j, and the output vector C_i' is obtained from the attention weight. The calculation for the attention layer is as follows:

$$e_{ij} = g\left(h_{i-1}, h_j\right) \tag{2}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} = \operatorname{softmax}(e_{ij}) \tag{3}$$

$$C_i' = \sum_{j=1}^{t} \alpha_{ij} h_j \tag{4}$$

where α_{ij} represents the attention weight, and e_{ij} represents the attention relevance score from time step j to time step i. The nonlinear function g is used to compare the hidden state h_j of the input sequence X_j at time step j (from the encoder) with the hidden state h_{i-1} of the generated sequence Y_i at the previous time step (from the decoder). It calculates the matching degree between the input sequence X_j and the generated sequence Y_i by referencing the datasets of each cluster. The higher the matching degree is, the larger e_{ij} and α_{ij} will be, which means that the output at time step i will allocate more attention to the input at time step j. Therefore, in this sub-model, the influence of time step j on time step i is greater. C_i' is the output vector of the attention layer at time step i, h_j is the hidden state at time step j, α_{ij} is the attention weight from time step j to time step i, and t is the time step of the input.

The C_i' obtained from the MSA module is used as the input sequence to the corresponding LSTM's decoding layer in the sub-model, where it is processed by the decoding layer to compute Y_i, as expressed by

$$\begin{cases} f_i = \delta(\omega_f \cdot [Y_{i-1}, h_i, C_i'] + b_f) \\ i_i = \delta(\omega_i \cdot [Y_{i-1}, h_i, C_i'] + b_i) \\ L_i = \tanh(\omega_C \cdot [Y_{i-1}, h_i, C_i'] + b_C) \\ C_i = f_i \cdot C_{i-1} + i_i \cdot \tanh(\omega_C \cdot [Y_{i-1}, h_i, C_i'] + b_C) \\ O_i = \delta(\omega_o \cdot [Y_{i-1}, h_i, C_i'] + b_o) \\ Y_i = O_i \cdot \tanh(C_i) \end{cases} \tag{5}$$

where f_i, i_i, and O_i represent the forget gate, input gate, and output gate, respectively. L_i and C_i represent the updates to the cell state, while C_{i-1} represents the cell state at time step $i-1$. δ is the sigmoid activation function. ω_f, ω_i, ω_C, and ω_o are the input weights, while b_f, b_i, b_C, and b_o are the corresponding bias terms. Y_{i-1} is the predicted output of the decoder at time step $i-1$, h_i is the hidden state of the decoder at time step i, and C_i' is the output vector of the attention layer at time step i.

When predicting the water discharge at time node ti, the attention mechanism of each sub-model increases the weight of the input sequence's time points, as shown in Figure 5. For the current time step's prediction, the attention module boosts the attention weights of the time points close to this time step. The attention module adjusts the attention weight for each time step based on the condition clusters of a single time node, considering the cluster to which it belongs and the distance from the time step ti. When predicting for the same time node ti, the output values $C'_{3,ti}$, $C'_{2,ti}$, $C'_{1,ti}$ from different attention modules are distinct from each other.

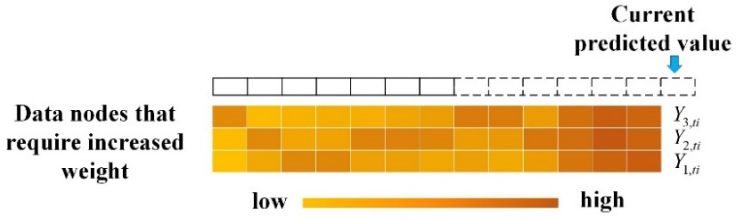


FIGURE 5. When predicting the water consumption at time node ti, the attention mechanism of each sub-model increases the weight of the input sequence's time points.

At the same time, different condition sub-models will simultaneously calculate different water consumption predictions, making it difficult to determine the accuracy of each sub-model's prediction based solely on the data at the current time. This paper designs a series of water consumption condition migration cost functions across continuous time nodes to optimize and find the most optimal condition sub-model's prediction at the current time. First, the proposed model defines a sliding window that includes a series of continuous time nodes and water consumption process variables. For each individual time node, the model calculates the water consumption predictions and operational condition membership degrees for three sub-models. Then, it defines the water consumption migration cost function between adjacent time nodes and the objective function for the optimal migration path of water consumption within the sliding window. Finally, the model utilizes a multi-source path optimization algorithm to solve the problem and records both the current water consumption prediction and the migration path of the predicted values within the sliding window.

3.3.2. *Water consumption condition migration matrix.* First, a dynamic sliding window is established to link the predicted water consumption at the current time with historical values. As shown in Figure 6, the interval between the most recent two water consumption validation values is used as the unit length of the sliding window, and the position of the sliding window moves forward dynamically with each update of the validation values. The condition migration matrix has a total of m columns, with each column corresponding to a time node of the sliding window. It represents the condition membership degree and the water consumption prediction value of the condition sub-model for the process variables at a given time.

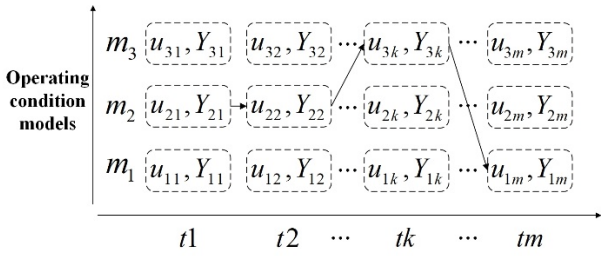


FIGURE 6. Condition network prediction process

The process of calculating the sub-model prediction value at time node ti is shown in Figure 7. The input time series is used to solve for the water consumption prediction value $Y = [Y_1, Y_2, Y_3]^{\mathrm{T}}$ corresponding to condition sub-model $m = [m_1, m_2, m_3]^{\mathrm{T}}$. Then, the corresponding cluster center $C = [C_1, C_2, C_3]^{\mathrm{T}}$ is identified, and the membership degree $u = [\mu_1, \mu_2, \mu_3]^{\mathrm{T}}$ of the prediction values of different condition sub-models at this node is calculated. Finally, the data $[(\mu_1, Y_1)_{ti}, (\mu_2, Y_2)_{ti}, (\mu_3, Y_3)_{ti}]^{\mathrm{T}}$ for this time node is

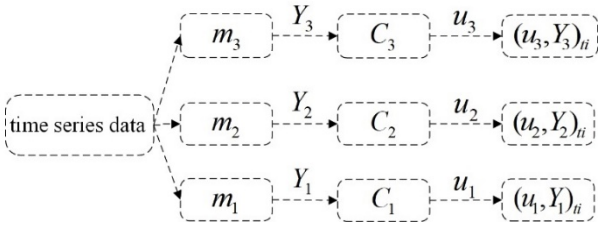


FIGURE 7. Sub-model prediction values at time node ti and membership degree to condition clusters

obtained. $(\mu_1, Y_1)_{ti}$ represents the membership degree of the process variable at time node ti to the first condition cluster and the sub-model prediction value.

3.3.3. *Optimal migration path for water consumption prediction values.* The migration path is set to fully utilize the predicted values of different condition sub-models while integrating with the dynamic conditions of water consumption. Considering the short-term stability of actual water usage and the gradual variation of water consumption conditions, the optimal migration path for water consumption prediction values selects the smoothest path from the condition migration matrix. Each column in the condition migration diagram corresponds to a specific column in the condition migration matrix. The current condition model can migrate to any condition model at the next time node. A complete condition migration path starts from the source node in the first column of the condition migration diagram and continues until the terminal node in the last column.

The migration connections between adjacent time node condition sub-model predictions in the condition migration diagram are shown in Figure 8. The migration cost function for water consumption conditions between adjacent time nodes is defined as

$$f(x_{p(i-1),qi}) = \left[\left(\frac{1}{\mu_{p(i-1)}} + \frac{1}{\mu_{qi}}\right)\left|x_{p(i-1)} - x_{qi}\right|\right]^2 \tag{6}$$

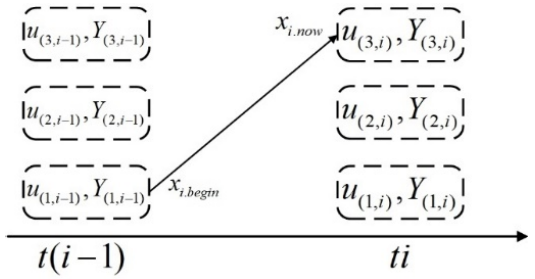


FIGURE 8. Migration cost function for condition sub-models between adjacent time nodes

In the formula, $f(x_{p(i-1),qi})$ is the cost function for migrating from the p-th condition sub-model prediction value at node $i-1$ to the q-th condition sub-model prediction value at node i. $x_{p(i-1)}$ is the prediction value of the p-th condition sub-model, and $u_{p(i-1)}$ is the membership degree of the p-th condition sub-model prediction value at node $i-1$.

The optimal water consumption prediction value at the current time can be obtained using the condition migration matrix. The objective function for the optimal migration path of water consumption is defined as

$$cost = \min\left(\sum_{i=2}^{m}\left(f\left(x_{p(i-1),qi}\right)\right)\right)$$
$$\text{s.t.}\begin{cases} f\left(x_{p(i-1),qi}\right) = \left[\left(\dfrac{1}{\mu_{p(i-1)}} + \dfrac{1}{\mu_{qi}}\right)\left|x_{p(i-1)} - x_{qi}\right|\right]^2 \\ \mu_1 + \mu_2 + \mu_3 = 1 \\ 1 \le p \le 3;\ 1 \le q \le 3 \\ \text{if } (i = t_a) \Rightarrow Y_i = Y_a \end{cases} \tag{7}$$

In the formula, m represents the number of time nodes included in the condition migration matrix, i.e., the number of time steps to be predicted. t_a denotes the validation time node, where the water consumption Y_a at this moment is set as the corresponding newly

added validation value of water consumption, which is used as the benchmark for path optimization.

3.3.4. *Multi-source path optimization algorithm.* Use the multi-source path optimization algorithm to solve the optimal water consumption operating condition sub-model prediction values. First, construct the adjacency matrix and calculate the minimum cost migration path between adjacent time nodes. The adjacency matrix for the water consumption prediction values of two adjacent time nodes is

$$A_{ij} = \begin{bmatrix} f(x_{i1,j1}), f(x_{i1,j2}), f(x_{i1,j3}) \\ f(x_{i2,j1}), f(x_{i2,j2}), f(x_{i2,j3}) \\ f(x_{i3,j1}), f(x_{i3,j2}), f(x_{i3,j3}) \end{bmatrix} \tag{8}$$

In the formula, time node i is adjacent to node j, and $f(x_{i1,j3})$ is the migration cost function from the first sub-model at node i to the third sub-model at node j. The minimum migration cost for the prediction value of the third operating condition sub-model from node i to node j is

$$\min_{j3}^{i} = \min\left[f(x_{i1,j3}), f(x_{i2,j3}), f(x_{i3,j3})\right] \tag{9}$$

Record the water consumption prediction values corresponding to each node, and ultimately obtain the complete migration path. Define $\min_{jn}^{i}$ $(n = 1, 2, 3)$ as the minimum migration cost between two adjacent time nodes, and $x_{jn}^{\min}$ $(n = 1, 2, 3)$ as the local optimal water consumption prediction value at node j when the migration cost is minimized. Define the memory matrix $\boldsymbol{R}_j$ as

$$\boldsymbol{R}_j = \begin{bmatrix} \min_{j1}^{i}, & x_{j1}^{\min}, & cost_{j1} \\ \min_{j2}^{i}, & x_{j2}^{\min}, & cost_{j2} \\ \min_{j3}^{i}, & x_{j3}^{\min}, & cost_{j3} \end{bmatrix} \tag{10}$$

In the formula, the first column represents the minimum migration cost of each sub-model prediction value from node i to node j; the second column shows the prediction results of the three sub-models at node j under the minimum migration cost; the third column is the cumulative value of the minimum migration cost for each sub-model prediction value from the source node to node j. The steps of the multi-source path optimization algorithm are as follows.

Step 1: Specify the source node i as the initial node, initialize the minimum cost matrix $[cost_{j1}, cost_{j2}, cost_{j3}]^{\mathrm{T}}$ with all values set to 0, indicating that the shortest path distance from the source node to itself is 0.

Step 2: Use the adjacency matrix of node i and the next adjacent time node j to calculate the minimum migration cost $\left[\min_{j1}^{i}, \min_{j2}^{i}, \min_{j3}^{i}\right]^{\mathrm{T}}$ for the three sub-models from node i to node j, as well as the prediction values of the sub-models at node j under the minimum migration cost $\left[x_{j1}^{\min}, x_{j2}^{\min}, x_{j3}^{\min}\right]^{\mathrm{T}}$. Then, calculate the sum $[cost_{j1}, cost_{j2}, cost_{j3}]^{\mathrm{T}}$ of the minimum migration costs for the sub-model prediction values from the source node to node j, and combine these three column vectors to form the memory matrix $\boldsymbol{R}_j$.

Step 3: If node j is the endpoint of the operating condition migration matrix, proceed to Step 4. If there are more nodes after node j, perform $i = j$, $j = j + 1$ and then return to Step 2.

Step 4: Upon reaching the terminal node j, stop the search, select the global shortest path to obtain the global minimum migration cost $cost_{\min} = \min[cost_{j1}, cost_{j2}, cost_{j3}]^{\mathrm{T}}$, and set the water usage prediction value calculated at the terminal node as the current node value $x_{\mathrm{si.now}}$. The optimal sub-model's water consumption prediction value at the current moment is set to $x_{\mathrm{si}} = x_{\mathrm{si.now}}$.

Step 5: Use the memory matrix $\boldsymbol{R}_j$ to search for the previous node $x_{\text{si.begin}}$ of $x_{\text{si.now}}$ and record the data of that node. Push this data onto stack $\boldsymbol{P}$.

Step 6: Determine whether $x_{\text{si.begin}}$ is the source node of the operating condition migration matrix. If the source node has not been reached, set $x_{\text{si.begin}}$ as the new current data point $x_{\text{si.now}}$, and return to Step 5. If the source node is reached, end the search. The data points in stack $\boldsymbol{P}$, when connected, form the shortest path, and the cumulative migration cost of the shortest path is $cost_{\min}$.

Algorithm. Multi-source path optimization

Input: Time-series $X = (X_1, X_2, X_3, \ldots, X_t)$, Time-step m
Output: Shortest path, x_{si}

Set the source node i, initialize $[cost_{i1}, cost_{i2}, cost_{i3}]^{\mathrm{T}}$
Calculate $\left[\min_{j1}^{i}, \min_{j2}^{i}, \min_{j3}^{i}\right]^{\mathrm{T}}$ and $\left[x_{j1}^{\min}, x_{j2}^{\min}, x_{j3}^{\min}\right]^{\mathrm{T}}$
$\boldsymbol{R}_j = [cost_{j1}, cost_{j2}, cost_{j3}]^{\mathrm{T}}$
If node j is the terminal node of the matrix:
    Go to 9
else:
    $i = j$
    $j = j + 1$
    back to 2
$cost_{\min} = \min [cost_{j1}, cost_{j2}, cost_{j3}]^{\mathrm{T}}$
$x_{\text{si}} = x_{\text{si.now}}$
Use the memory matrix $\boldsymbol{R}_j$ to search for the previous node $x_{\text{si.begin}}$ of $x_{\text{si.now}}$
Record the data of that node $x_{\text{si.begin}}$, which is pushed onto the stack $\boldsymbol{P}$
If $x_{\text{si.begin}}$ is the source node of the operating condition migration matrix:
    Go to 9
else:
    Set $x_{\text{si.begin}}$ as the new current data point $x_{\text{si.now}}$
    back to 11
The data points in stack $\boldsymbol{P}$ are connected to form the shortest path
The migration cost of the shortest path is accumulated as $cost_{\min}$
Return shortest path and x_{si}

Through the above steps, the optimal operating condition sub-model water consumption prediction path and the optimal water consumption prediction value x_{si} at the current moment are obtained from the operating condition migration matrix. In the optimization path, the optimal operating condition sub-model corresponding to each time node can be observed, which migrates the water consumption prediction values between adjacent time nodes as smoothly as possible. This fits the variation characteristics of the actual water consumption process, reducing the impact of noise from process variables in the water consumption flow on the prediction accuracy of the operating condition network.

3.3.5. *Fully connected layer.* Multilayer Perceptron (MLP) consists of multiple fully connected layers, with an input layer, hidden layers, and an output layer, where neurons between layers are connected in a fully connected manner. The proposed model uses MLP as the fully connected layer to integrate spatial features deepening the network's depth and enhancing its learning capability during training.

The prediction output Y_{ti} at time ti is obtained from the stack $\boldsymbol{P}$ output, and then the Y_{ti} is calculated through the leaky ReLU activation function of the MLP module, resulting in the final predicted sequence $Y(Y_{t1}, Y_{t2}, Y_{t3}, \ldots, Y_{tm})$.

4. Experiment and Results Analysis.

4.1. Experimental dataset and experimental settings.

The hardware environment of the experimental platform is a desktop computer with an Intel i5-12600KF 4.9GHz CPU, 16GBx2 memory, and a Gigabyte RTX 4060 Ti 16GB graphics card. The software environment of the experimental platform includes MATLAB 2023b and PyCharm 2022 Community Edition.

The specific training steps are as follows. 1) Perform factor analysis on the collected multi-variable water consumption dataset to obtain a reduced feature set, which serves as the new feature variable dataset. 2) Apply K-means++ clustering on the feature set to forming three clusters. Each cluster's data points represent the operational conditions corresponding to that cluster. 3) Insert an MSA module between the encoder and decoder layers of three LSTM models. The data points corresponding to each operational condition in the time dimension are input into the MSA module of the respective sub-model. 4) The proposed model generates the prediction results for each of the three operational condition sub-models at a single time node. 5) The membership degrees of the three current prediction results are calculated, and the minimum-cost migration path is identified using the operational condition network. The prediction results output from the stack are then passed through the MLP fully connected layer for final output.

The proposed model is trained using four-fold cross-validation on the training set. First, the training set is evenly divided into four mutually exclusive subsets (fold1, fold2, fold3, fold4), each with approximately equal time lengths. The mini-batch gradient descent algorithm is used for iteration. Each fold is sequentially used as the validation subset, with the remaining three folds used as the training subsets. This process completes four rounds of training and validation, and the validation errors for each fold are calculated and averaged. The best set of parameters is selected based on the lowest validation error as the training parameters for the training set.

The mini-batch gradient descent algorithm combines the advantages of both batch gradient descent and stochastic gradient descent. In each iteration, a mini-batch of training samples is used to calculate the gradient. During training, the initial learning rate is set to 0.002. If the model's performance does not improve during the training process, the learning rate is reduced by 0.0001. The loss function is set to Mean Squared Error (MSE), as shown in the following expression:

$$\mathrm{E} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{11}$$

In the formula, N is the total number of input samples for a single sub-model; y_i is the true value of the i-th sample; $\hat{y}_i$ is the predicted value of the i-th sample.

4.2. The selection of experimental model.

In order to accurately evaluate the performance of the LSTM-MSA network model proposed in this paper, comparative experiments are conducted on the model designed in this paper, the reproduced Transformer [21], LSTM-GRU [20], CNN-LSTM [29], LSTM-RNN [30], and LSTM-attention-LSTM [22] models.

When training the model with the water consumption dataset, the Sigmoid function is used as the activation function. The parameters for testing are obtained through grid search within the parameter search range specified in the respective papers.

4.3. Experiment valuation indicators.

Three commonly used performance evaluation metrics in time series forecasting are selected: Mean Absolute Percentage Error (MAPE),

Root Mean Square Error (RMSE), and Coefficient of Determination (R^2). Their specific formulas are as follows:

$$\text{MAPE} = \frac{100\%}{T}\sum_{i=1}^{T}\left|\frac{Y_{pre,i} - Y_{real,i}}{Y_{real,i}}\right| \tag{12}$$

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}\left(Y_{real,i} - Y_{pre,i}\right)^2} \tag{13}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{T}\left(Y_{real,i} - Y_{pre,i}\right)^2}{\sum_{i=1}^{T}\left(Y_{real,i} - \bar{Y}\right)^2} \tag{14}$$

where $Y_{real,i}$ is the true value of the i-th data point in the sequence, $Y_{pre,i}$ is the predicted value of the i-th data point in the sequence, and T is the length of the sequence. The smaller the values of MAPE and RMSE are, the closer the predicted values are to the true values, indicating better model performance. The closer R^2 to 1, the better the model's fit; conversely, the closer to 0, the poorer the model's fit.

4.4. **Experimental evaluation.** This section evaluates the performance of each model on datasets A, B, and C under conditions of missing data values, insufficient feature quantity, and different prediction time steps. The experimental results of each model on 10 random test subsets of datasets A, B, and C are averaged to obtain the final MAPE, RMSE, and R^2.

4.4.1. *Experiment evaluation of missing data.* Experiments were conducted on dataset A to predict with the LSTM-MSA operational condition network designed in this study, as well as with Transformer and LSTM-based models, using a time interval of 60 minutes and a prediction horizon of 24. As shown in Figure 10, in dataset A, which contains a significant amount of missing data, most models exhibit overfitting when the prediction horizon reaches 14. This is due to the use of mean imputation from neighboring data when data is missing. As the prediction horizon increases beyond a certain point, overfitting becomes unavoidable. When the prediction horizon is less than 14, the proposed model demonstrates better robustness by leveraging operating condition migration network. This is further evident from the MAPE box plots of the models shown in Figure 9.

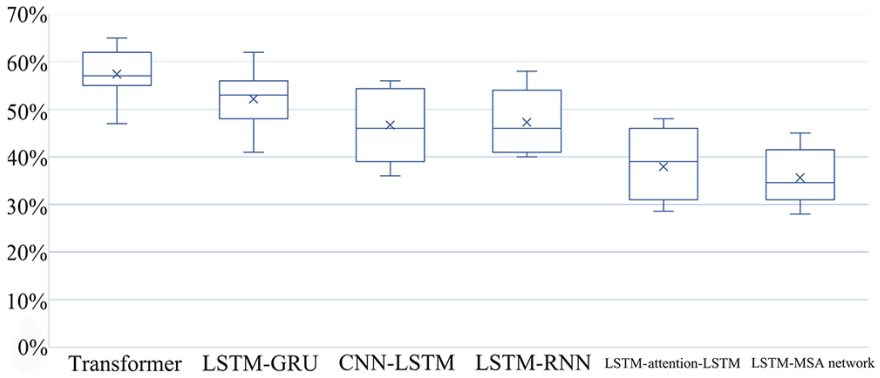


FIGURE 9. The MAPE boxplot of each model on dataset A

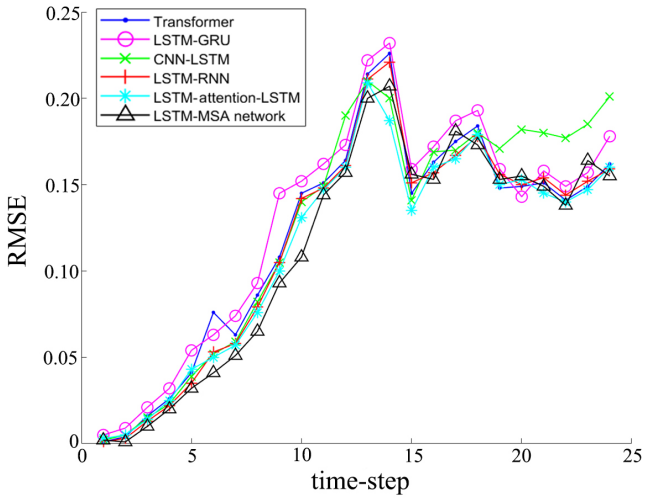


FIGURE 10. The RMSE line chart of each model on dataset A

TABLE 1. Prediction evaluation results of each model on dataset A

Model	Transformer			LSTM-GRU			CNN-LSTM			LSTM-RNN			LSTM-attention-LSTM			LSTM-MSA network		
	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2
Data A	58.6%	0.162	0.79	51.9%	0.178	0.81	47.8%	0.201	0.87	48.2%	0.159	0.80	39.1%	0.160	0.85	35.5%	0.155	0.92

4.4.2. *Experiment evaluation of insufficient feature.* Experiments were conducted on dataset B to predict using the LSTM-MSA operational condition network designed in this study, as well as Transformer and LSTM-based models, with a time interval of 15 minutes and a prediction horizon of 24. As shown in Figure 12, when the instantaneous flow rate of the groundwater inlet feature is missing in dataset B, the performance of the proposed model is almost unaffected, whereas the reference LSTM-based models are impacted to varying degrees due to the missing data dimensions. This is most evident in the LSTM-RNN model, which requires high data completeness for its construction, as reflected in its R^2 value in Table 2. The proposed LSTM-MSA model outperforms the LSTM-attention-LSTM model in terms of RMSE for most time steps, as the data preprocessing method involving factor analysis and clustering improves the performance of the MSA module under conditions of insufficient features.

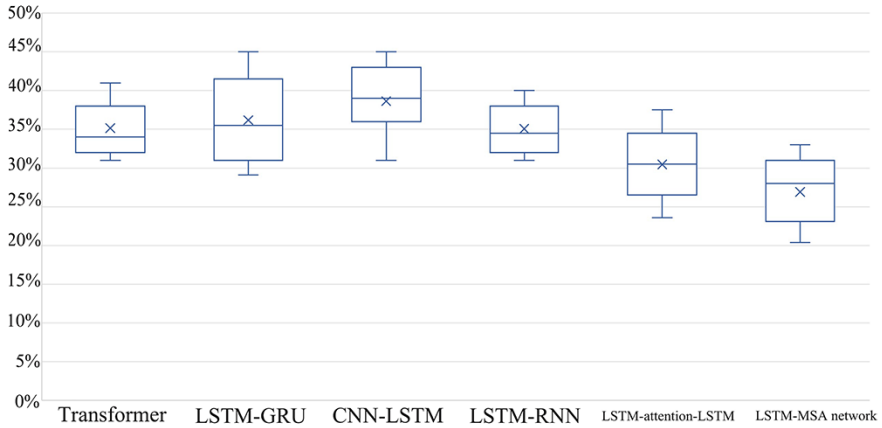


FIGURE 11. The MAPE boxplot of each model on dataset B

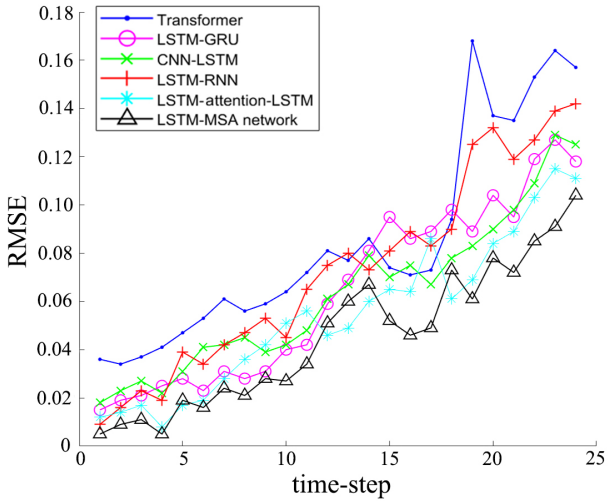


FIGURE 12. The RMSE line chart of each model on dataset B

TABLE 2. Prediction evaluation results of each model on dataset B

Model	Transformer			LSTM-GRU			CNN-LSTM			LSTM-RNN			LSTM-attention-LSTM			LSTM-MSA network		
	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2
Data B	35.1%	0.157	0.86	35.9%	0.118	0.87	38.5%	0.142	0.85	34.9%	0.125	0.84	30.2%	0.111	0.90	26.8%	0.104	0.94

4.4.3. *Experiment evaluation of short-term and long-term prediction.* Experiments were conducted on dataset C to predict using the LSTM-MSA operational condition network designed in this study, as well as Transformer and LSTM-based models, with a time interval of 5 minutes and a prediction horizon of 24. As shown in Table 3, the proposed model achieves the smallest RMSE and MAPE values, and the largest R^2, indicating that the model outperforms other models in terms of prediction accuracy when the data is complete and features are sufficient. The proposed model also retains an advantage over LSTM-GRU, CNN-LSTM, LSTM-RNN, and LSTM-attention-LSTM. The Transformer model excels in short-term predictions, but due to its large output horizon, it performs poorly on all three metrics. The following analysis will compare the prediction capabilities of the proposed model with the Transformer in the short-term and with LSTM-based models in the long-term, based on prediction curves.

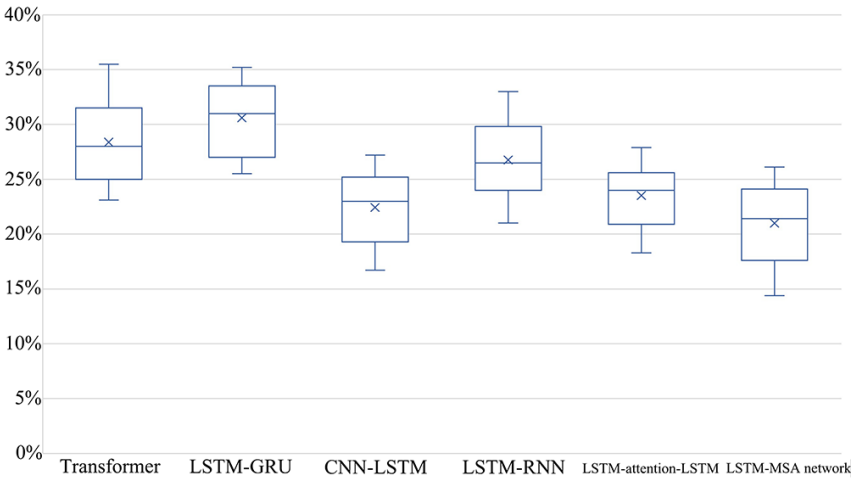


FIGURE 13. The MAPE boxplot of each model on dataset C

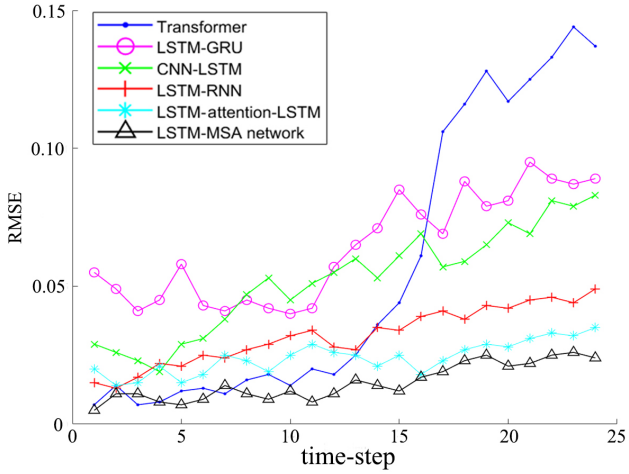


FIGURE 14. The RMSE line chart of each model on dataset C

TABLE 3. Prediction evaluation results of each model on dataset C

Model	Transformer			LSTM-GRU			CNN-LSTM			LSTM-RNN			LSTM-attention-LSTM			LSTM-MSA network		
	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2	MAPE	RMSE	R^2
Data C	28.5%	0.137	0.74	30.8%	0.089	0.87	22.3%	0.083	0.84	26.3%	0.049	0.93	24.2%	0.035	0.92	20.9%	0.024	0.95

4.4.3.1. *Short-term prediction capability evaluation.* Experiments were conducted on dataset C to predict using the LSTM-MSA operational condition network and the Transformer model, with a time interval of 5 minutes and a prediction horizon of 12. The actual water consumption values over a 24-hour period, along with the predictions from the two models, are shown in Figure 15. As can be seen in Figure 14, the Transformer model, which incorporates an attention mechanism, excels at capturing data characteristics from shorter sequences, leading to strong performance in short-term predictions under the same operational conditions. However, as the prediction horizon increases, the Transformer model, which only considers a single operational condition, shows an increase in RMSE when there is a change in the operational condition. This is evident in the latter part of the prediction curve in Figure 15. In contrast, the LSTM-MSA operational condition network accounts for the presence of different operational conditions in advance

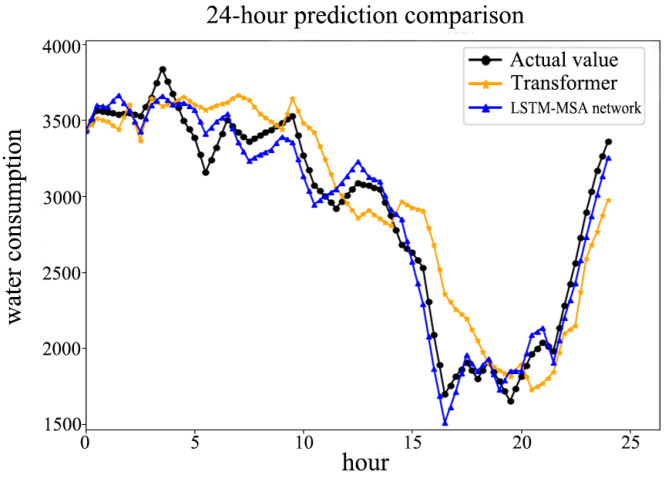


FIGURE 15. LSTM-MSA network and Transformer prediction curves for 24 hours

by constructing sub-models for each condition. This approach helps prevent delays in the attention mechanism during predictions. Since the sub-models are built based on datasets obtained through factor analysis and clustering, they emphasize the decision-making attributes of the cumulative effects of feature variables on water consumption predictions.

4.4.3.2. *Long-term prediction capability evaluation.* Experiments were conducted on dataset C to predict using the LSTM-MSA operational condition network, LSTM-GRU, and LSTM-attention-LSTM models, with a time interval of 5 minutes and a prediction horizon of 24. The actual water consumption values over a 72-hour period, along with the predictions from the three models, are shown in Figure 16. The MAPE in Figure 13 and the RMSE in Figure 14 indicate that the method proposed in this paper inherits the strength of LSTM in long-term sequence prediction. As shown in Table 3 and Figure 16, the LSTM-MSA operational condition network achieves the highest degree of fit. The LSTM-GRU model determines the optimal parameter configuration through performance comparison experiments based on the number of hidden layers. However, it overlooks the fact that, as the prediction horizon increases, the model tends to forget previous data. The LSTM-attention-LSTM model incorporates an attention mechanism into the LSTM, providing a clear advantage for predicting larger time series with longer horizons. However, it does not account for ensuring smooth transitions between predictions, resulting in a noticeable decline in the goodness-of-fit as the prediction horizon increases. The proposed LSTM-MSA operational condition network addresses these issues by integrating an attention mechanism and matrix-based multi-source path optimization within the LSTM. This approach enhances the model's predictive accuracy in complex environments, meeting both accuracy and robustness requirements for general scenarios.

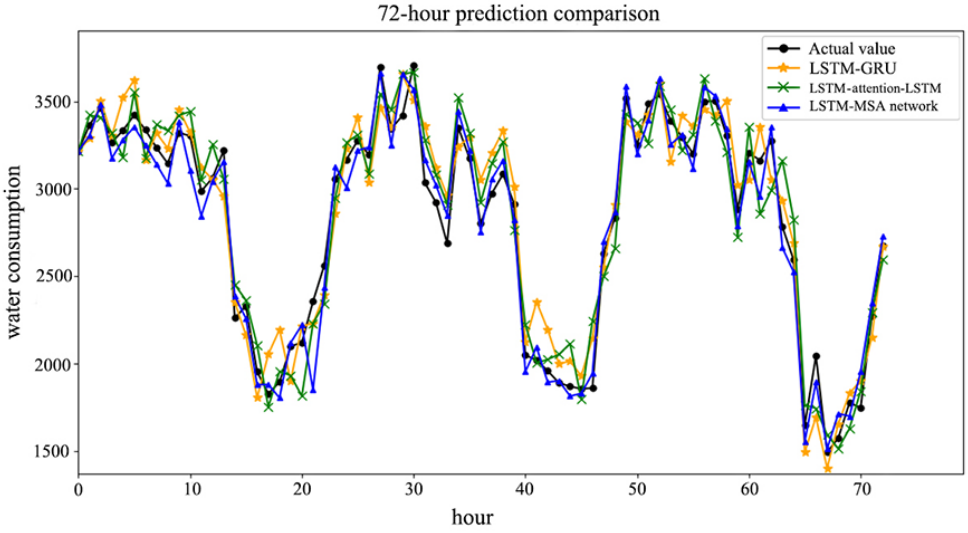


FIGURE 16. Prediction curves for 72 hours using LSTM-based models

5. **Conclusions.** A model based on LSTM-MSA network is suggested to address challenges in multi-dimensional water consumption time series forecasting, such as the large number of relevant features, non-stationary feature fluctuations, and variations in operational conditions during the water consumption process. Firstly, the proposed model performs factor analysis for dimensionality reduction on the collected multivariate time series, followed by K-means++ clustering. This process quantifies the correlation between multidimensional data and addresses the issue of insufficient feature extraction of operational conditions, resulting in multi-condition divisions and visualizations. Secondly, the

proposed model incorporates a multi-head self-attention mechanism into the LSTM to construct predictive sub-models for the corresponding operational condition clusters. The influence of multiple factors on individual time steps may lead to variations in the operational condition categories across different time points. To address this issue, the proposed model enhances the weight of data points within the corresponding cluster of each operational condition sub-model. Finally, the proposed model incorporates a multi-source path optimization algorithm for operational condition migration, which integrates information from both the time and spatial domains. This algorithm determines the optimal operational condition migration path for predicting water consumption at the current time step, effectively fitting the operational condition change process in actual production. The proposed method is tested on the publicly available datasets from Samuel Street – South 9th Street, Brooklyn, and the self-constructed dataset from a water utility company in Shenyang, considering the impacts of factors such as missing data values, insufficient feature numbers, and different prediction horizons. Experimental results demonstrate that the proposed method outperforms similar forecasting approaches, demonstrating higher accuracy and robustness in general scenarios.

To further enhance the practicality of the proposed method, the following issues need to be addressed in the next phase of work: 1) The proposed algorithm should incorporate iteration into the K-means++ clustering algorithm to reduce the displacement of cluster center positions caused by ambiguous points in the initial values, avoiding the risk of falling into local optima during clustering. Improving the accuracy of the cluster centers can indirectly reduce the deviation between the predicted and actual values of the model's operating condition network; 2) The proposed algorithm should improve the intra-cluster similarity in cases where there are a significant number of missing sample values, thereby enhancing the rationality of the attention module's allocation of attention in the prediction sub-model; 3) The proposed algorithm has high hardware requirements, and consideration should be given to how to reasonably modify the model structure to further improve computational efficiency under the constraint of limited computational resources in practical production environments.

REFERENCES

[1] Y. Cao, Z. Wang, P. Li et al., Prediction of rural domestic water and sewage production based on automated machine learning in northern China, *Journal of Cleaner Production*, vol.434, 140016, 2024.

[2] S. D. Latif and A. N. Ahmed, Streamflow prediction utilizing deep learning and machine learning algorithms for sustainable water supply management, *Water Resources Management*, vol.37, no.8, pp.3227-3241, 2023.

[3] J. Zheng and M. Huang, Traffic flow forecast through time series analysis based on deep learning, *IEEE Access*, vol.8, pp.82562-82570, DOI: 10.1109/ACCESS.2020.2990738, 2020.

[4] U. M. Sirisha, M. C. Belavagi and G. Attigeri, Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison, *IEEE Access*, vol.10, pp.124715-124727, DOI: 10.1109/ACCESS.2022.3224938 2022.

[5] Z. Jiang, C. Xu, W. Gui et al., Prediction method of hot metal silicon content in blast furnace based on optimal smelting condition migration, *Acta Automatica Sinica*, vol.48, no.1, pp.194-206, 2022.

[6] Q. Liu, H. Peng, L. Long et al., Nonlinear spiking neural systems with autapses for predicting chaotic time series, *IEEE Transactions on Cybernetics*, vol.54, no.3, pp.1841-1853, DOI: 10.1109/TCYB.2023.3270873, 2023.

[7] L. Chisci, A. Mavino, G. Perferi et al., Real-time epileptic seizure prediction using AR models and support vector machines, *IEEE Transactions on Biomedical Engineering*, vol.57, no.5, pp.1124-1132, DOI: 10.1109/TBME.2009.2038990, 2010.

[8] J. L. Torres, A. Garcia, M. De Blas et al., Forecast of hourly average wind speed with ARMA models in Navarre (Spain), *Solar Energy*, vol.79, no.1, pp.65-77, 2005.

[9] Y. Wang, D. Wang and Y. Tang, Clustered hybrid wind power prediction model based on ARMA, PSO-SVM, and clustering methods, *IEEE Access*, vol.8, pp.17071-17079, DOI: 10.1109/ACCESS.2020.2968390, 2020.

[10] D. Lee, D. Lee, M. Choi et al., Prediction of network throughput using ARIMA, *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp.1-5, DOI: 10.1109/ICAIIC48513.2020.9065083, 2020.

[11] S. Zhang, Y. Wang, M. Liu et al., Data-based line trip fault prediction in power systems using LSTM networks and SVM, *IEEE Access*, vol.6, pp.7675-7686, DOI: 10.1109/ACCESS.2017.2785763, 2018.

[12] W. Deng, K. Li and H. Zhao, A flight arrival time prediction method based on cluster clustering-based modular with deep neural network, *IEEE Transactions on Intelligent Transportation Systems*, DOI: 10.1109/TITS.2023.3338251 2023.

[13] D. A. Tedjopurnomo, Z. Bao, B. Zheng et al., A survey on modern deep neural network for traffic prediction: Trends, methods and challenges, *IEEE Transactions on Knowledge and Data Engineering*, vol.34, no.4, pp.1544-1561, DOI: 10.1109/TKDE.2020.3001195, 2020.

[14] J. Siłka, M. Wieczorek and M. Woźniak, Recurrent neural network model for high-speed train vibration prediction from time series, *Neural Computing and Applications*, vol.34, no.16, pp.13305-13318, 2022.

[15] J. Schmidhuber and S. Hochreiter, Long short-term memory, *Neural Comput.*, vol.9, no.8, pp.1735-1780, 1997.

[16] L. Li, W. Xie and X. Zhou, Cooperative spectrum sensing based on LSTM-CNN combination network in cognitive radio system, *IEEE Access*, vol.11, pp.87615-87625, DOI: 10.1109/ACCESS.2023.3305483, 2023.

[17] S. Xiang, J. Zhou, J. Luo et al., Cocktail LSTM and its application into machine remaining useful life prediction, *IEEE/ASME Transactions on Mechatronics*, vol.28, no.5, pp.2425-2436, DOI: 10.1109/TMECH.2023.3244282, 2023.

[18] E. Koo and G. Kim, A hybrid prediction model integrating GARCH models with a distribution manipulation strategy based on LSTM networks for stock market volatility, *IEEE Access*, vol.10, pp.34743-34754, DOI: 10.1109/ACCESS.2022.3163723, 2022.

[19] X. Wan, H. Liu, H. Xu et al., Network traffic prediction based on LSTM and transfer learning, *IEEE Access*, vol.10, pp.86181-86190, DOI: 10.1109/ACCESS.2022.3199372, 2022.

[20] M. Cho, C. Kim, K. Jung et al., Water level prediction model applying a long short-term memory (LSTM)-gated recurrent unit (GRU) method for flood prediction, *Water*, vol.14, no.14, 2221, 2022.

[21] A. Ghosh, Time series transformer for long term rainfall forecasting towards water distribution management in smart cities, *IEEE International Conference on Big Data (BigData)*, pp.3380-3386, DOI: 10.1109/BigData59044.2023.10386081, 2023.

[22] X. Wen and W. Li, Time series prediction based on LSTM-attention-LSTM model, *IEEE Access*, vol.11, pp.48322-48331, DOI: 10.1109/ACCESS.2023.3276628, 2023.

[23] A. Niknam, H. K. Zare, H. Hosseininasab et al., Developing an LSTM model to forecast the monthly water consumption according to the effects of the climatic factors in Yazd, Iran, *Journal of Engineering Research*, vol.11, no.1, 100028, 2023.

[24] M. El Hanjri, H. Kabbaj, A. Kobbane et al., Federated learning for water consumption forecasting in smart cities, *IEEE International Conference on Communications (ICC 2023)*, pp.1798-1803, DOI: 10.1109/ICC45041.2023.10279576, 2023.

[25] A. A. Nasser, M. Z. Rashad and S. E. Hussein, A two-layer water demand prediction system in urban areas based on micro-services and LSTM neural networks, *IEEE Access*, vol.8, pp.147647-147661, DOI: 10.1109/ACCESS.2020.3015655, 2020.

[26] W. Sun and C. Huang, Predictions of carbon emission intensity based on factor analysis and an improved extreme learning machine from the perspective of carbon emission efficiency, *Journal of Cleaner Production*, vol.338, 130414, 2022.

[27] *New York Water Consumption and Cost*, https://www.kaggle.com/datasets/odezi45/new-york-water-consumption-and-cost, Accessed on Jul. 5, 2024.

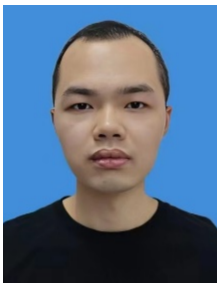
[28] *Water Consumption and Costs 2013-Feb 2023*, https://www.kaggle.com/datasets/usmanlovescode/water-consumption-and-costs-2013-feb-2023, Accessed on Jul. 5, 2024.

[29] Z. Pu, J. Yan, L. Chen et al., A hybrid wavelet-CNN-LSTM deep learning model for short-term urban water demand forecasting, *Frontiers of Environmental Science & Engineering*, vol.17, no.2, 22, 2023.

[30] H. Kardhana, J. R. Valerian, F. I. W. Rohmat et al., Improving Jakarta's katulampa barrage extreme water level prediction using satellite-based Long Short-Term Memory (LSTM) neural networks, *Water*, vol.14, no.9, 1469, 2022.

Author Biography

Fenglong Kan received the B.E. degree in Automation Engineering from Shenyang Jianzhu University, China, in 2004, the M.E. degree in Testing Technology from Shenyang Jianzhu University, China, in 2010. He is currently an Associate Professor and a Supervisor of master student with Shenyang Jianzhu University, China. His current research interests include machine learning, and multi-attribute decision-making.



Weiyang Li received the B.E. degree in Building Electrical and Intelligence from Shenyang Jianzhu University, China, in 2019. He is currently pursuing the M.E. degree in Control Science and Engineering at Shenyang Jianzhu University, China. His current research interests include machine learning, and adaptive switching control.

Guowei Yuan received the B.E. degree in Mechanical Design and Manufacturing from Jiamusi University, China, in 2002, the M.E. degree in Mechanical Design and Manufacturing from Dalian Maritime University, China, in 2016. He is currently an Associate Professor at the Liaoning Mechanical & Electrical College of Technology, China. His current research interests include mechanical design optimization, intelligent manufacturing systems, and precision machining technologies.