Hindawi
Structural Control and Health Monitoring
Volume 2023, Article ID 2189912, 19 pages
https://doi.org/10.1155/2023/2189912

Research Article

MLA-TCN: Multioutput Prediction of Dam Displacement Based on Temporal Convolutional Network with Attention Mechanism

Yu Wang and Guohua Liu

College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China

Correspondence should be addressed to Guohua Liu; zjuliugh@zju.edu.cn

Received 8 December 2022; Revised 25 June 2023; Accepted 2 August 2023; Published 25 August 2023

Academic Editor: Tzu-Kang Lin

The displacement of concrete dams effectively reflects their structural integrity and operational status. Therefore, establishing a model for predicting the displacement of concrete dams and studying the evolution mechanism of dam displacement is essential for monitoring the structural safety of dams. Current data-driven models utilize artificial data that cannot reflect the actual status of dams for network training. They also have difficulty extracting the temporal patterns from long-term dependencies and obtaining the interactions between the targets and variables. To address such problems, we propose a novel model for predicting the displacement of dams based on the temporal convolutional network (TCN) with the attention mechanism and multioutput regression branches, named MLA-TCN (where MLA is multioutput model with attention mechanism). The attention mechanism implements information screening and weight distribution based on the importance of the input variables. The TCN extracts long-term temporal information using the dilated causal convolutional network and residual connection, and the multioutput regression branch achieves simultaneous multitarget prediction by establishing multiple regression tasks. Finally, the applicability of the proposed model is demonstrated using data on a concrete gravity dam within 14 years, and its accuracy is validated by comparing it with seven state-of-the-art benchmarks. The results show that the MLA-TCN model, with a mean absolute error (MAE) of 0.05 mm, a root-mean-square error (RMSE) of 0.07 mm, and a coefficient of determination (R^2) of 0.99, has a comparably high predictive capability and outperforms the benchmarks, providing an accurate and effective method to estimate the displacement of dams.

1. Introduction

Dam health monitoring is important for ensuring the safety of dams. The monitoring process involves collecting and analyzing data to detect any structural issues or safety concerns. Dam monitoring models are widely developed for inspecting dam health and can predict the operational status of dams by establishing a structural response under the combined influence of external factors, such as reservoir pressure, water temperature, and air temperature, and internal factors, such as dam material properties, e.g., creep, and alkali-aggregate reactions. As an important indicator of dam health monitoring, the displacement of dams can effectively reflect the operational status and structural integrity of the dam [1, 2]. Therefore, it is essential to establish a model for predicting dam displacement and study the evolution of displacement to monitor and maintain the structural safety of dams.

1.1. Overview of Research Background. Modern dam monitoring models can be classified into machine-learning and deep-learning models. Machine-learning models [3–9] use machine-learning algorithms to carry out nonlinear regression analysis between dam response and influencing factors, thereby establishing a mapping between the two. For instance, Su et al. [10] introduced a prediction method for dam structural responses based on the support vector machine (SVM) model, demonstrating enhanced accuracy over conventional SVM models. Dai et al. [11] suggested an optimized statistical model utilizing random forest regression for concrete dam deformation monitoring, showing superior prediction capabilities for dam displacements compared to traditional statistical models. Salazar et al. [12] used boosted regression trees for identifying and assessing dam deformation and leakage, with their evaluation results surpassing traditional statistical analysis methods. Kao et al. [13] explored the monitoring of long-term static deformation data of the Fei-Tsui arch dam using artificial neural network (ANN)-based techniques. Ren et al. [14] achieved precise dam displacement prediction by integrating grey relational analysis, random forest, and the backpropagation (BP) neural network. These models solve multicollinearity problems in traditional statistical methods and significantly improve accuracy and computational speed compared to traditional methods [15]. However, models based on machine learning cannot extract the temporal patterns from dependencies and do not apply to dam displacement prediction with time effects.

In the recent five years, new displacement prediction models leveraging deep-learning algorithms, such as long short-term memory network (LSTM) and gated recurrent units (GRU), have been devised due to their exceptional performance across diverse fields. These models possess memory and display advantages when learning the nonlinear characteristics of time series data, making them more suitable for predicting time-dependent factors like dam displacement. For instance, Liu et al. [16] proposed a coupled model for predicting the long-term displacement of arch dams based on an LSTM network, which incorporates principal component analysis and the moving average method. Qu et al. [17] designed a single-point and multipoint model for estimating concrete dam displacement, using the rough set theory and an LSTM network. Shu et al. [18] introduced a model using the variational autoencoder and the temporal attention-based LSTM network for predicting long-term arch dam displacement. Bui et al. [19], aiming to optimize the LSTM for highly accurate dam displacement prediction, designed a model based on the LSTM, combined with the coronavirus optimization algorithm, for estimating the displacement of a hydropower dam. Wen et al. [20] proposed a prediction model based on a fusion of multiresolution analysis (MR) and a stacked GRU neural network. Yang et al. [21] proposed a technique that combines complementary ensemble empirical mode decomposition (CEEMD) and GRU for predicting the displacement of reservoir landslides.

While deep-learning-based dam displacement prediction models have significantly improved and demonstrated superior performance over machine-learning-based models, practical application challenges persist.

(1) These approaches use empirical formulas to construct input samples, which do not reflect the actual operational status of the dam and make it harder to train the model

(2) Recurrent architectures (such as LSTM and GRU) struggle to learn long-term patterns from a longer history sequence length so that they cannot perform well in dam displacement prediction

(3) Current data-driven methods are inconclusive about the progression and relationship between dam displacement and environmental variables, which is critical to exploring the dam displacement evolution and physical mechanisms

1.2. Innovations and Contributions. As previously mentioned, deep-learning-based dam displacement prediction models have shortcomings that necessitate further research. Dam displacement evolution is a dynamic process, with researchers aiming to establish a relationship between dam displacement and influencing factors to allow for automatic and real-time identification of the dam's operational status. However, models built on artificial data, created via empirical formulas, may not accurately represent the dam's actual condition. To develop a reliable mapping, it is essential to use raw monitoring data as the direct input for modeling both displacement and environmental factors.

The progression of dam displacement is a slow process, typically taking several months or more, necessitating a model capable of capturing its long-term sequential evolution nature, with strong temporal processing capabilities. While recurrent architectures can learn long-term temporal dependencies, they are hampered by limited short-term memory, making it challenging to learn long-term patterns from longer historical sequences. Researchers typically apply LSTM models with a time series length of approximately 10 in applications [19, 22]. However, recurrent architectures may be inadequate for dam displacement prediction due to their short-term memory limitations. Alternatively, the temporal convolutional network (TCN) can extract useful features from longer sequence lengths, making it a potentially effective solution for extracting temporal information in dam displacement prediction. Bai et al. [23] introduced TCN in 2018, combining the parallel feature processing capacity of convolutional neural networks with the sequence modeling capabilities of recurrent neural networks. It outperforms baseline recurrent architectures in various sequence modeling tasks. Although TCN has been successfully implemented in numerous fields such as engineering life prediction [24], industrial load analysis [25], and meteorological prediction [26], its application in dam health monitoring remains limited. As far as we know, this study is the first to apply TCN to dam displacement prediction.

Dam displacement evolution is affected by multiple variables, such as time, water level, and temperature, each with varying levels of impact. Understanding the relationships between input variables and output results helps to establish temporal evolution laws and interactions between variables. Thus, it is essential for the displacement prediction model to reflect the relationship between displacement and these variables. A standalone TCN model might not suffice for dam displacement prediction. To address this, the attention mechanism [27] can be used to focus the model on critical information, improving its interpretability. Earlier works have explored the use of attention mechanisms in neural networks, such as Cao et al.'s method of combining TCN and residual self-attention for predicting the remaining useful life of rolling bearings. However, their work only used the self-attention mechanism to enhance the model's performance without analyzing the relationship between features and the target variable. Yang et al. [28] also designed a model to estimate concrete dam displacement using the attention mechanism and the LSTM network. Ren et al. [29] furthered this model by incorporating both temporal and factor attention, suggesting an interpretable model that combines the attention mechanism and LSTM networks. Although previous studies have paired attention mechanisms with LSTM networks to enhance LSTM's interpretability, they did not examine the process of input variable changes. This study concentrates on effectively merging the attention mechanism with TCN and examining the effect of inverse input variables on dam displacement.

There are multiple sensors installed inside the dam; thus, using a single-point displacement model is not suitable. It is required to retrain the network when predicting the displacement of multiple points, which not only increases the calculation complexity but also disregards the correlation between points [30]. Therefore, it is necessary to use multipoint displacement modeling to improve the generalization of the model.

Accordingly, this study proposes a multioutput model based on the temporal convolutional network and with the attention mechanism for predicting the displacement of concrete dams. The model regards environmental variables as the input and the displacement of a dam as the output. It predicts the displacement of the dam by establishing the relationship between the displacement and the environmental variables. The primary contributions of this work are as follows:

(1) The proposed model uses raw monitored data as input to establish a real relationship between dam displacement and environmental variables

(2) The proposed model utilizes TCN to learn long-term features from a longer history sequence length and introduces the attention mechanism to help TCN focus on important information and take multioutput regression branches to predict multiple targets in parallel, reflecting the dam displacement evolution process as realistically as possible

(3) The proposed model explores the process of how input variables affect the displacement changes by visualizing the attention weights

The rest of this study is structured as follows. Section 2 introduces the theoretical foundation and algorithm background, and Section 3 describes the implementation details of the proposed MLA-TCN method. Section 4 presents a case study and the analysis of the results, and Section 5 demonstrates a comparative analysis of the developed model. Finally, Section 6 provides conclusions and future perspectives.

2. Fundamental Theory

This section explains the theoretical basis of the proposed MLA-TCN model, including its three aspects: the attention mechanism, the temporal convolutional network, and the multioutput regression analysis.

2.1. Attention Mechanism. The attention mechanism is a particular architecture embedded in machine-learning models to automatically learn the contribution of the input to the output. It has a profound optimization effect on conventional models. The attention mechanism adopts an encoder-decoder architecture, which traditionally transfers the final hidden state from the encoder to the decoder. In contrast, the attention mechanism delivers all the outputs from the encoder to the decoder, preventing losing practical information in the encoder-decoder periods of long-term sequences. In addition, visualizing the attention mechanism can enhance the interpretability of the model.

This work utilizes the global attentional model Luong et al. proposed in 2015 for feature screening and weight assignment [31]. The approach takes an alignment model receiving the outputs of all encoders as the input and exports a score by calculating the dot product of the encoder input and the hidden states of the decoder, all of which are passed through a location-based function to obtain the terminal weights of the output of each encoder. The formulas are defined as follows:

$$\tilde{h}_{(t)} = \sum_{i} \alpha_{(t,i)} y_{(i)}, \tag{1}$$

$$\alpha_{(t,i)} = \frac{\exp\left(e_{(t,i)}\right)}{\sum_{i'} \exp\left(e_{(t,t')}\right)}, \tag{2}$$

$$e_{(t,i)} = \begin{cases} h_{(t)}^{T} y_{(i)}, \\ h_{(t)}^{T} W y_{(i)}, \\ v^{T} \tan h\left(W\left[h_{(t)}; y_{(i)}\right]\right). \end{cases} \tag{3}$$

At each time step (t), the model infers a variable-length alignment weight vector, i.e., $\alpha_{(t,i)}$, based on the current target state, $h_{(t)}$, and all source states, $\tilde{h}_{(t)}$.

2.2. Temporal Convolutional Network. The TCN is a powerful toolkit for sequence modeling. It introduces a one-dimensional convolutional neural network and causal convolution to produce an output of the same length as the input, and there can be no leakage from the future into the past. Meanwhile, it employs the dilated convolution network and residual connection to achieve long-term temporal feature extraction. Causal convolution predicts the current output from past input data. If we take one-dimensional time series $X = (x_0, x_1, \ldots x_t, \ldots, x_T) \epsilon R^n$ as the input, equation (4) expresses the causal convolution operation, $F(\cdot)$, for sequence element t. The TCN also introduces dilation convolution in addition to causal convolution to perform long history sequence assignments [32]. Equation (5) expresses the dilated convolution of sequence element t computing $F(\cdot)$ for a one-dimensional sequence input $X = (x_0, x_1, \ldots x_t, \ldots, x_T) \epsilon R^n$ and filter $F = (f_1, f_2, \ldots, f_K)$.

Furthermore, the temporal convolutional network introduces a generic residual block as an alternative to the convolutional layer to increase the network depth for expanding the receptive field [33]. Within a residual block, the TCN has two layers of dilated causal convolution, weight normalization, activation layer, dropout layer, and one-dimensional convolution. Equation (6) defines the output of the ith residual block, $Z(i)$:

$$(F * X)_{(x_t)} = \sum_{k=1}^{k} f_k \cdot x_{t-K+k}, \tag{4}$$

$$\left(F *_d X\right)_{(x_t)} = \sum_{k=1}^{k} f_k \cdot x_{t-(K-k)\bullet d}, \tag{5}$$

$$Z^{(i)} = \delta\left(F\left(Z^{(i-1)}\right) + Z^{(i-1)}\right), \tag{6}$$

where d represents the dilation factor, K is the number of filters, $t - d\cdot(K - k)$ indicates the past direction, and $\delta(\cdot)$ denotes the activation operation.

Figure 1 illustrates the architectural elements of a temporal convolutional network, and Figure 1(a) depicts the causal convolutional architecture. For input $X = (x_0, x_1, \ldots x_t, \ldots, x_T)$, output y at time t only depends on the inputs from the current time (x_t) and from the partial past time but not on any future inputs. Figure 1(b) describes the dilated causal convolution with a filter size (K) of 2 and dilation factors (d) of 1, 2, and 4, which can perform a longer history sequence than the causal convolution in Figure 1(a). Figure 1(c) details the generic residual block, where one branch performs a transformation operation on the input, while the other conducts a simple 1 × 1 convolution transformation to maintain the consistency of the number of feature maps in parallel with the existing branch.

2.3. Multioutput Regression Analysis.

Most existing methods for predicting the displacement of dams adopt single-output regression analysis, which can only address regression problems involving a single target variable. Dam displacement involves multiple measurement points and directions, which are correlated, so they cannot be regarded as entirely independent. Therefore, single-output regression analysis cannot satisfy the requirements of dam displacement prediction.

Thus, this work utilizes multioutput regression analysis in constructing the MLA-TCN model to address these problems, which can deal with regression problems involving multiple variables and improve the generalization of the model. The major difference between multioutput regression and single-output regression is the number of labels. The process of multioutput regression is illustrated in Figure 2. For features $X = (X_0, X_1, \ldots, X_N) \in R^n$ and labels $Y = (Y_1, Y_2, \ldots, Y_M)$, multioutput regression analysis establishes an equation H: $X \longrightarrow Y$ between X and Y based on the sample set $D = \{(X, Y_i) | 1 < i \leq m\}$. The loss function of the multioutput regression analysis is the superposition of each target loss function, and the total loss is the sum of the differences between the estimated and actual values of each target.

3. Methodology

This study devises a novel method based on the attention mechanism and the temporal convolutional network to predict the displacement of dams by establishing a relationship between dam displacement and environmental variables. Figure 3 presents the overall framework of the proposed MLA-TCN model. The proposed model comprises three parts: (1) data acquisition and preprocessing, (2) network construction and training, and (3) model prediction and evaluation.

The first step is to acquire monitored data using monitoring equipment such as sensors installed in the dam body, preprocess the obtained data with various methods, and finally make them into a dataset.

Next, the network is constructed using an encoder-decoder architecture, incorporating an attention mechanism in the encoder to weigh the significance of the data, a temporal convolutional network in the encoder for long-term temporal information processing, and multioutput regression analysis in the decoder for predicting multiple targets at once. The model is trained using the Huber loss function, and hyperparameters are tuned using the Hyperband algorithm.

Finally, we utilize the pretrained MLA-TCN model to predict dam displacement, and the performance of the model is estimated by absolute mean error (MAE), root-mean-square error (RSME), and the coefficient of determination (R^2). The following detail the MLA-TCN method.

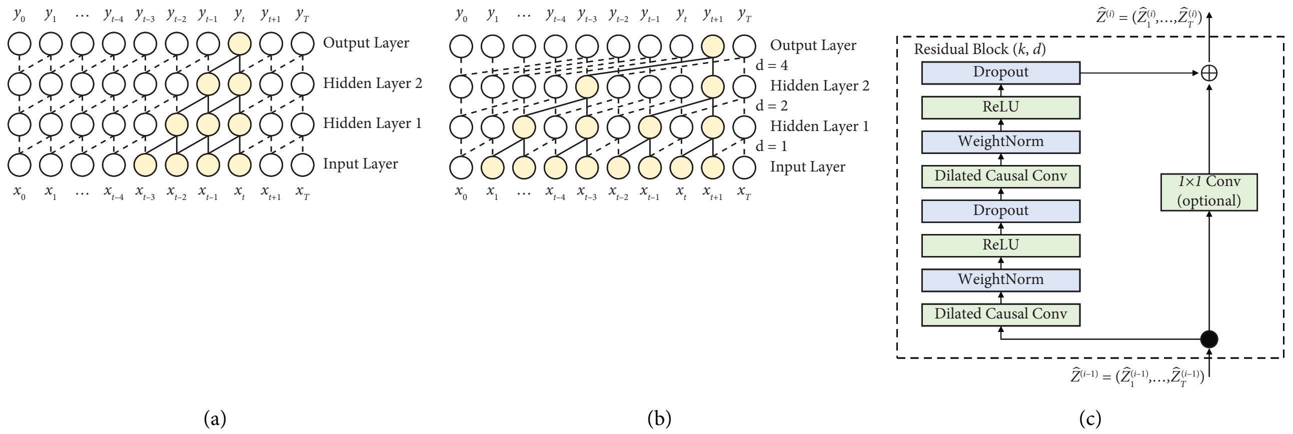
3.1. Data Acquisition and Preprocessing.

The MLA-TCN method utilizes experimental monitored data for modeling. Generally, monitored data are obtained from equipment, such as sensors, preset inside the dam, which are subject to the abnormality, absence, and inconsistent data format caused by environmental disturbance and equipment corruption. Feeding such raw data directly into the neural network without processing misleads model predictions, so these raw data require operations such as outlier processing, missing value processing, and normalization processing. The trained samples are not representative, so it is essential to set untrained samples to test the model for analysis, and the whole dataset should be divided into a training set and a testing set.

3.2. Model Development.

This section intends to establish a structural response between dam displacement and environmental variables so as to predict dam displacement using environmental variables. The proposed model is constructed with an encoder-decoder architecture.

In the encoder, the attention mechanism layer is first introduced to process the input, as shown in Figure 4(a). For environmental variables $X = [x_1, x_2, \ldots, x_N]$, the input features are $X = [x_{1,T}, x_{2,T}, \ldots, x_{N,T}]$ when the length of the temporal sequence is T. The attention mechanism layer calculates the weight of each input variable for each time step (t) employing procedure ① and then computes the weight ratio of each input variable using procedure ②. Finally, it



y_0 y_1 … y_{t-4} y_{t-3} y_{t-2} y_{t-1} y_t y_{t+1} y_T
Output Layer
Hidden Layer 2
Hidden Layer 1
Input Layer
x_0 x_1 … x_{t-4} x_{t-3} x_{t-2} x_{t-1} x_t x_{t+1} x_T
d = 4
d = 2
d = 1
$\hat{Z}^{(i)} = (\hat{Z}_1^{(i)},\ldots,\hat{Z}_T^{(i)})$
Residual Block (k, d)
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1×1 Conv (optional)
$\hat{Z}^{(i-1)} = (\hat{Z}_1^{(i-1)},\ldots,\hat{Z}_T^{(i-1)})$
(a)
(b)
(c)

Figure 1: The architectural elements in a temporal convolutional network: (a) a causal convolution with a filter size of 2; (b) a dilated causal convolution with dilation factors of 1, 2, and 4 and a filter size of 2; (c) the TCN residual block. A 1 × 1 convolution is added when the residual input and output have different dimensions.

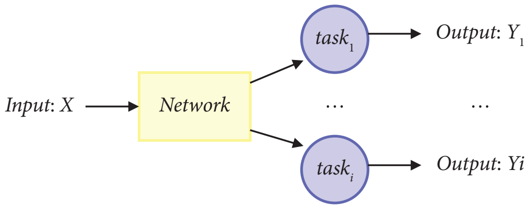
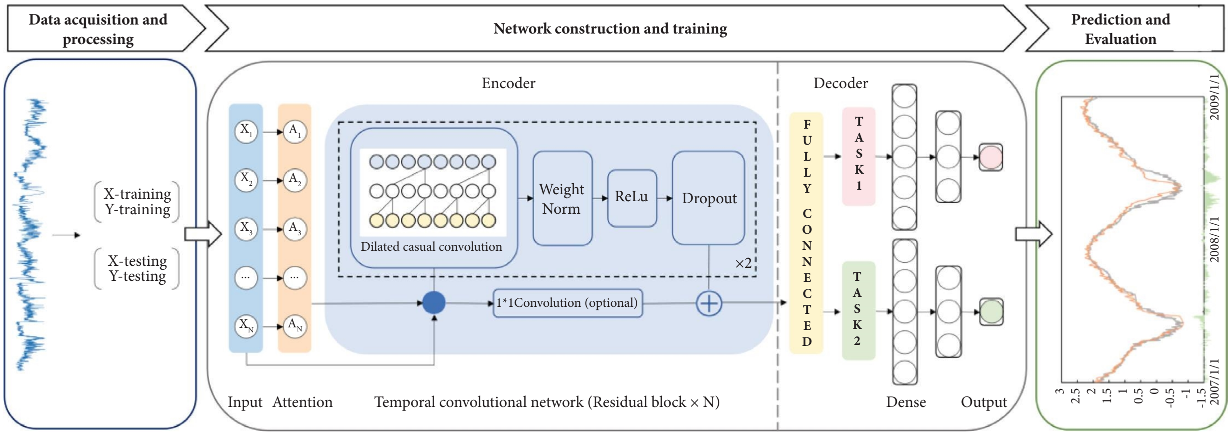


Figure 2: The multioutput regression process diagram.



Data acquisition and processing
Network construction and training
Prediction and Evaluation
X-training
Y-training
X-testing
Y-testing
Encoder
Decoder
X_1 X_2 X_3 … X_N
A_1 A_2 A_3 … A_N
Dilated casual convolution
Weight Norm
ReLu
Dropout
×2
1*1Convolution (optional)
FULLY CONNECTED
TASK 1
TASK 2
Input
Attention
Temporal convolutional network (Residual block × N)
Dense
Output
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
2007/1/1
2008/1/1
2009/1/1

Figure 3: The overall framework of the designed MLA-TCN method.

obtains the weight assignments employing procedure ③, merged with the source features to compose the output of the attention mechanism layer. Equations (1)–(3) express steps ①, ②, and ③, respectively. The process does not compress the input variables, and each input has a corresponding weight to guarantee data integrity, as shown in Figure 4(b).

Afterward, a deep temporal convolutional network is used to extract information from the weight-assigned features. It consists of multiple generic temporal convolutional network residual blocks in series, as shown in Figure 5(a). Each block has a dilated causal convolution layer, a weight normalization layer, an activation layer, a dropout layer, and a one-dimensional convolution layer in series. The dilated causal convolution layer performs long-term temporal feature extraction. The weight normalization layer limits the weight range to increase training speed, while the activation layer maps the output nonlinearly using rectified linear units. Finally, the dropout layer randomly abandons some convolution work during training to prevent overfitting.

Furthermore, the TCN residual block employs an additional 1 × 1 convolution to ensure that the element addition receives a tensor of the same shape so as to account for the difference in the widths of the input and output.

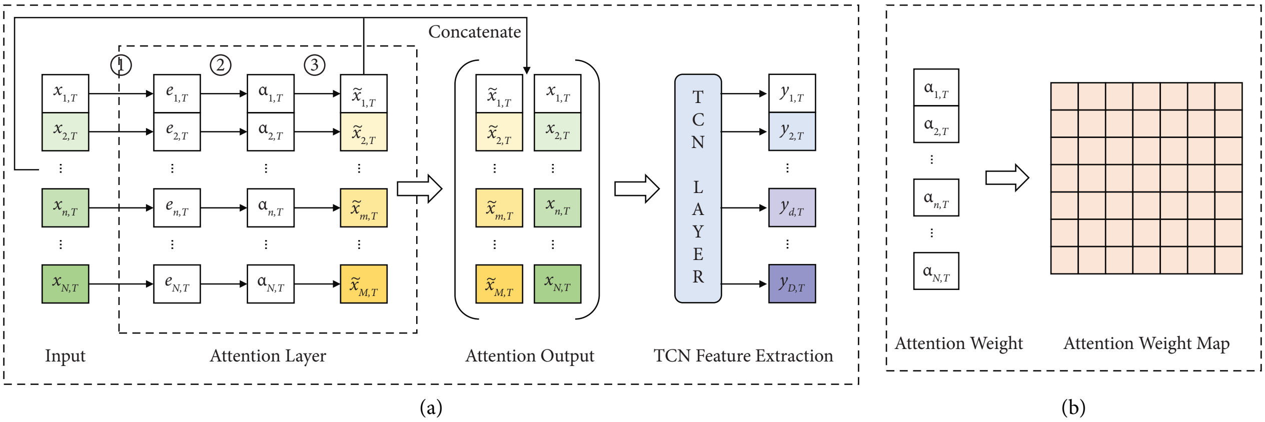


Figure 4: The architecture of the attention mechanism: (a) the attention mechanism based on factor screening, where ① indicates the calculation of the factor weights, ② denotes the softmax of the factor weights, and ③ represents the result of the factor screening; (b) the visualization of the attention weights.

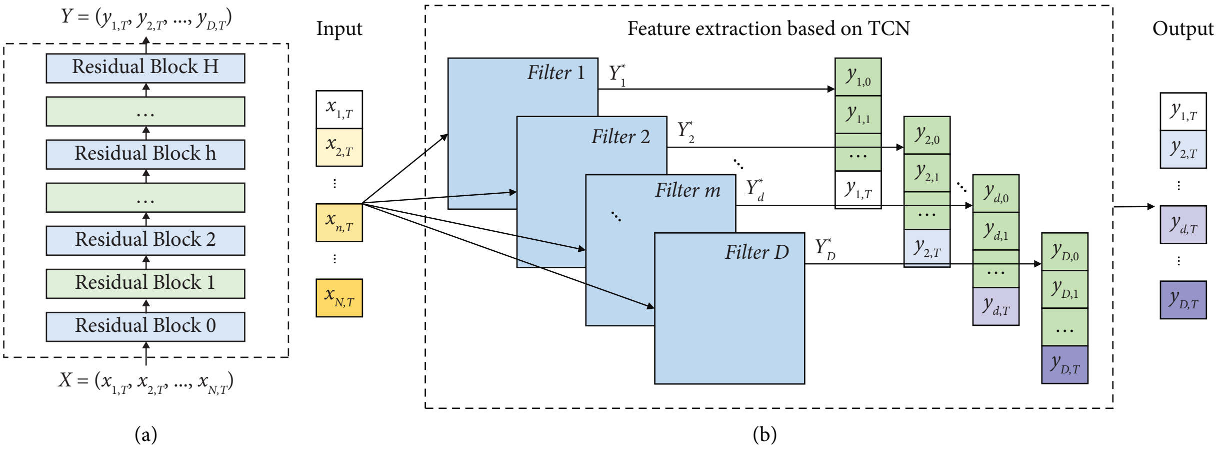


Figure 5: The TCN-based feature extraction network: (a) the deep temporal convolutional network; (b) the feature extraction processes based on the TCN.

Figure 5(b) illustrates the feature extraction process of a long sequence based on the deep TCN. The outputs of the attention mechanism layer $X = [X_{1,T}, X_{2,T}, \ldots, X_{N,T}]$ are fed into a deep temporal convolutional network and computed over several TCN residual blocks, where D convolutional kernels are designed for convolutional operations to obtain D sequences, i.e., $Y_1^*, Y_2^*, \ldots, Y_d^*, \ldots, Y_D^*$; the dth tensor can be expressed by $Y_d^* = (Y_{d,0}, Y_{d,1}, \ldots, Y_{d,T})$, and finally, $Y_d^* = (Y_{d,0}, Y_{d,1}, \ldots, Y_{d,T})$ is obtained as the output of the temporal convolution layer.

In the decoder, the model introduces several branches to analyze multiple targets simultaneously. Each branch has the same architecture: a dropout layer, a batch normalization layer, and a fully connected layer. The batch normalization layer homogenizes the output of the convolutional layer to prevent gradient explosion and gradient disappearance. The fully connected layer integrates the local information of the convolutional layer through the weight matrix for regression analysis so as to finally obtain multiple target outputs, Y.

3.3. Model Training and Optimizing. The proposed model needs to be trained before making any predictions, and this is performed by minimizing the loss function. The MLA-TCN model has multiple branches with the same architecture and shared weights, so a corresponding loss function is designed to learn the relative features. As each branch has the same construction and task, the model's loss function is the same as that of a single branch. As expressed in equation (7), we adopt the Huber loss function [34] as the loss function for each branch. The Huber loss function is a multistage function that is used to calculate the loss for data with periodic variations and significant discrepancies. It uses mean square error when the difference between predicted and measured values is within the range of a hyperparameter ε and mean absolute error when it is outside the range of ε. This approach overcomes the limitations of single-stage loss functions that are affected by target value. The proposed model's loss function is designed to train the multibranched MLA-TCN model, and it is expressed in equation (8) as the superimposed loss between multiple predicted and actual values.

$$L_{\varepsilon} = \min_{\omega,\sigma} \sum_{i=1}^{n} \left(\sigma + H_{\epsilon}\left(\frac{X_i\omega - y_i}{\sigma}\right)\sigma \right) + \alpha\|\omega\|_2^{\ 2}, \tag{7}$$

where

$$H_{\epsilon}(z) = \begin{cases} z^2, & \text{if } |z| < \varepsilon, \\ 2\varepsilon|z| - \varepsilon^2, & \text{otherwise,} \end{cases} \tag{8}$$

$$L = \sum_{k=1}^{m} L_{\varepsilon}, \tag{9}$$

where y_i represents the ith real value, X_i denotes the ith input, n is the number of samples, ω and σ indicate the network parameters, H_{ε} stands for the Huber formula, ε is the hyperparameter, z represents the difference between the predicted and measured values of the network, and k denotes the number of multioutput regression targets.

For optimization objectives, the MLA-TCN model should be trained to find optimal parameters ω and σ. The MLA-TCN model is a feedforward neural network that utilizes the backpropagation algorithm to calculate its parameters. The network parameters are optimized using the small batch random gradient descent algorithm and the Adam algorithm to minimize the objective function, i.e., $L(\omega, \sigma)$.

3.4. Model Prediction and Evaluation. This study introduces three statistical metrics, namely, the mean absolute error, the root-mean-square error, and the coefficient of determination, to evaluate the performance of the MLA-TCN model. These evaluation metrics assess the model performance from multiple perspectives. The MAE measures the magnitude of the predicted value with respect to the measured value, while the RSME is sensitive to large-scale errors and accounts for the dispersion of the model; an RSME close to zero indicates accurate prediction. The coefficient of determination is utilized to measure the fitting of a sample regression line to data and ranges from 0 to 1.0, with a value close to 1.0 implying a perfect fit. The following formulas define the evaluation metrics:

$$\begin{aligned} \text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|, \\ \text{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|^2} \\ R^2 &= 1 - \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}, \end{aligned} \tag{10}$$

where n represents the number of samples, y_i denotes the ith real value, $\widehat{y}_i$ is the ith predicted value, and $\overline{y}$ indicates the average value.

4. Case Study

4.1. Overview of Hydropower Dam. This study uses an operational hydroelectric project as a case study. The hydropower station lies at the junction of the city of Baiyin and Yuzhong County, the province of Gansu, on the mainstream of the Yellow River, 65 km away from the main river of the city of Lanzhou in China, as illustrated in Figure 6(a). Figure 6(b) displays an overview of the upstream side of the project.

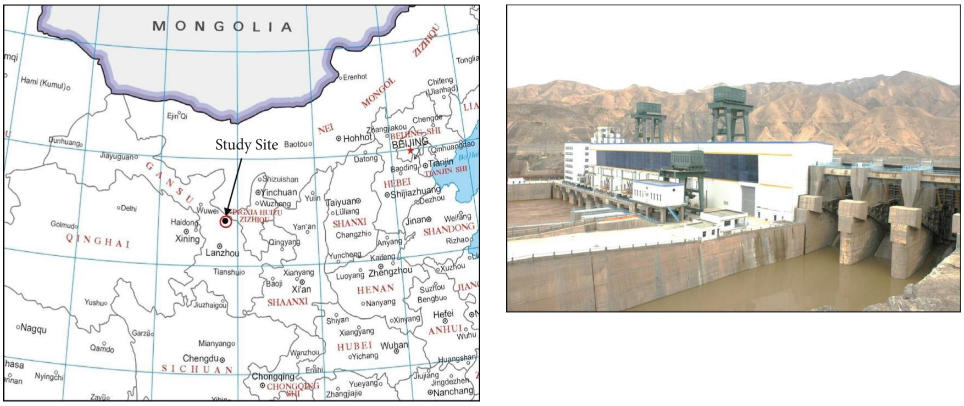
The dam's safety monitoring system was installed during construction, and an automated system was established in 1996. Various parameters are monitored, such as elevation control, environment volume, deformation, seepage, stress-strain response, and temperature. Water levels and temperature are measured both manually and automatically, while horizontal displacement is inspected using automatic and manual equipment.

Figure 7 shows the arrangement of the pendulum line automatic monitoring system. Three inverted pendulum lines were installed on the left dam abutment, the right auxiliary dam section, and the right dam abutment, numbered 1#, 6#, and 2#, respectively. Similarly, two inverted pendulum lines were placed on the operation gallery, numbered 4# and 3#, respectively, to check the terminal stability of the wire alignment. The 5# inverted pendulum line was positioned adjacent to the grouting gallery and the V-perpendicular line. A V-perpendicular line was equipped in the hydraulic turbine section 2#.

4.2. Data Acquisition and Preprocessing. The goal of this study is to predict the horizontal displacement of a dam using environmental variables, specifically the upstream water level, downstream water level, temperature, and time. Data from a 14-year period, from 2007 to 2020, are used for the analysis, and the upstream water level, downstream water level, and air temperature are considered as input features, and the specific measuring points, DC1, DC5, DC6(2), and ZC2, are used as prediction targets. Measuring points DC1 and DC5 represent the displacement of the dam foundation, and DC6(2) indicates the displacement of the dam crest. ZC2 measures the displacement of this position relative to the dam crest, which also indicates the displacement of the dam crest, as shown in Figure 7. The locations of the four specific measurement points are marked with ellipses in Figure 7.

Figure 8 illustrates the changes in the monitored data, where Figure 8(a) shows the variations in the upstream water level, downstream water level, and temperature, and Figure 8(b) displays the changes in the horizontal displacement of measuring point DC1, with X representing the relative displacement in the upstream-downstream direction and Y representing the relative displacement in the left and right bank directions. Table 1 summarizes the maximum, minimum, mean, and standard deviation values of the monitored data. The data from measuring points DC6 and ZC2 are relatively scattered, while the data from measuring points DC1 and DC5 are slightly discrepant.

Figure 8 shows deficiencies, outliers, and noise in the monitored data, so the raw data require preprocessing before constructing the dataset. We first replace the outliers with average values and then adopt the Lagrange interpolation method to fulfill the deficiency. Finally, we normalize the



(a) (b)

Figure 6: The hydropower dam: (a) the study site of the dam; (b) an overview of the dam.

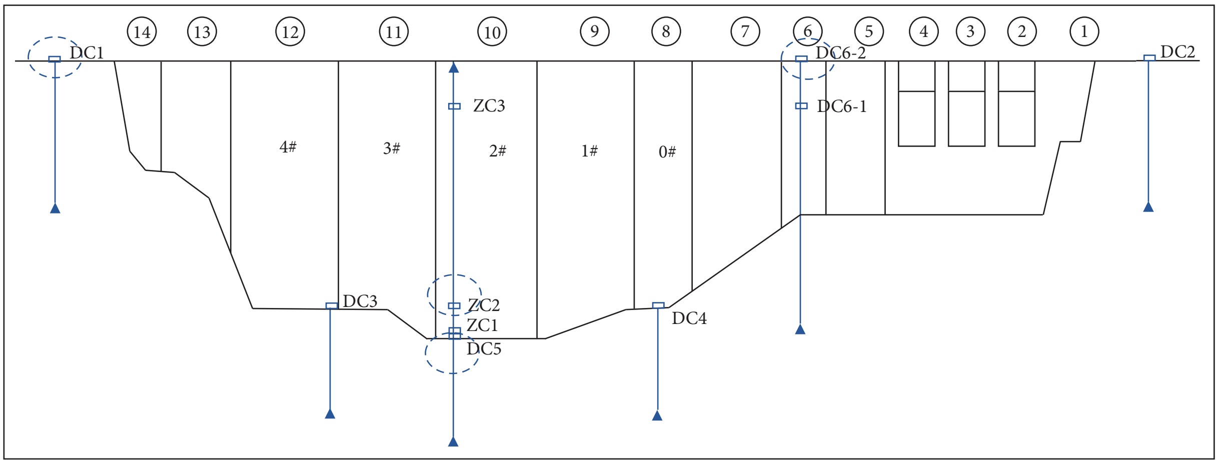
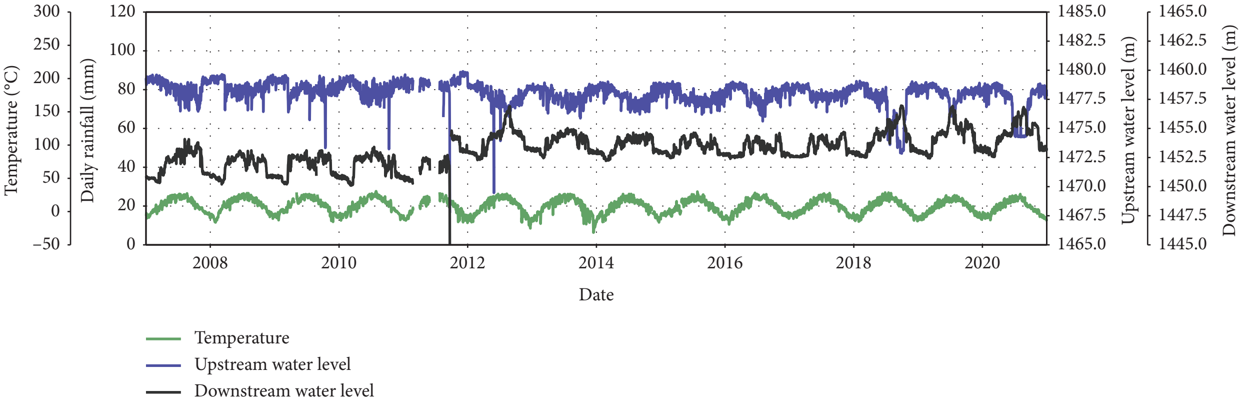
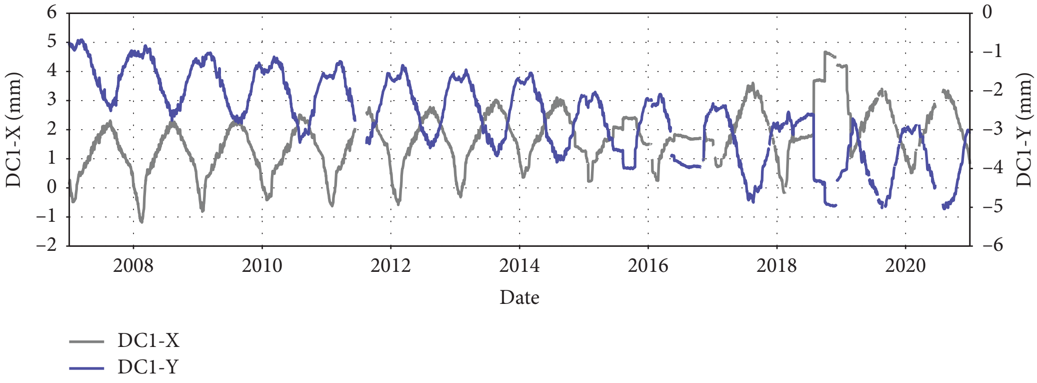


Figure 7: The arrangement of the pendulum line automatic monitoring system.



(a)

Figure 8: Continued.



(b)

Figure 8: The time-series monitored data: (a) the upstream water level, the downstream water level, and air temperature; (b) the horizontal displacement at measuring point DC1.

Table 1: The statistical analysis of the monitored data.

Variable	Unit	Maximum	Minimum	Mean	Standard deviation
DC1-X	mm	3.74	−0.84	1.67	0.96
DC1-Y	mm	−0.68	−5.04	−2.71	1.06
DC5-X	mm	−0.1	−2.54	−1.05	0.5
DC5-Y	mm	0.06	−2.19	−1.29	0.62
DC6-X	mm	6.04	−9.22	−1.77	2.56
DC6-Y	mm	6.08	−5.19	0.51	2.28
ZC2-X	mm	17.7	−10.45	3.58	4.13
ZC2-Y	mm	14.3	−13.92	1.8	5.07
Upstream water level	m	1480.32	1475.52	1478.09	0.80
Downstream water level	m	1456.61	1450.09	1453.14	1.32
Air temperature	°C	30.64	−29.10	7.99	11.33

processed data to a range of 0–1.0 using the following equation:

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{11}$$

where x_n represents the homogenized data, x denotes the input variable, $x_{\max}$ is the maximum value, and $x_{\min}$ indicates the minimum value.

After processing the data, we divide the data into the training set and the testing set with a ratio of 8 : 2. Subsequently, the datasets are fed into the model for training and optimization. To better assess the model's generalization ability and its performance on unknown data, we adopted the approach of using the first 80% of the data as the training set and the remaining 20% as the test set.

4.3. Environmental Settings and Parameter Configuration of Experiments. This section mainly presents the software and hardware environments and the corresponding parameter configuration of the MLA-TCN model. The proposed and comparison models were coded in Python 3.7 with the Spyder editor. All the models were executed utilizing a personal computer (PC) equipped with Intel® Core™ i7-8700K CPU operating at 3.70 GHz and NVIDIA GeForce RTX 2080 Ti, 11G.

To establish the MLA-TCN model for predicting the displacements of the dam, we require to reasonably determine four parameters, namely, the number of convolutional kernels in the TCN layer (KN), the size of convolutional kernels in the TCN layer (KS), the learning rate (LR), and the sequence length (SL), which significantly affect the model prediction performance. The hyperband algorithm [35], along with fivefold cross-validation, is used to find the best hyperparameter combinations (KN, KS, LR, and SL) for optimal model performance. Hyperband is an early stopping method based on successive halving, which aims to search for the best hyperparameters as quickly as possible. However, due to limited resources, not all hyperparameter combinations can be trained to convergence, so the successive halving algorithm is used to terminate some combinations early. This algorithm has two parameters, η and s, which control the proportion and number of hyperparameter combinations, respectively. The hyperband algorithm performs the successive halving algorithm multiple times with different s values and allocates the same resources to each one to find the best hyperparameter combination.

Based on the above algorithm, a four-dimensional search space (KN, KS, LR, and SL) is established. The hyperparameter ranges are set as follows: KN = [1, 16, 32, 64, 128, 256], KS ∈ [1, 9], LR = [0.01, 0.001, 0.0001], and SL = [10, 15, 100, 200, 400]. In the Hyperband algorithm, the number of iterative epochs is set at 30, and the batch size is set at 32. The hyperparameter combination of the MLA-TCN model determined by the Hyperband algorithm is as follows: KN = 64, KS = 3, LR = 0.01, and SL = 100.

4.4. Model Training and Validating. The optimized MLA-TCN model is employed for training, and the number of iteration epochs is set at 200 with the early stopping algorithm, which can avoid overfitting by terminating the training in advance. For each epoch, the loss and error of the training and testing sets are recorded. Figure 9(a) depicts the trend of the Huber loss of the MLA-TCN model during both the training and testing phases. The graph shows a gradual decrease in the Huber loss as the iteration increases and eventually stabilizes after 30 epochs. The optimal performance of the testing set is obtained after 176 epochs, which is when the iteration terminates, indicating a well-trained model without any overfitting or underfitting. Figures 9(b) and 9(c) demonstrate the variation in the MAE and RSME of the proposed MLA-TCN model during training and testing. The MAE and RSME follow the same trend as the Huber loss and gradually stabilize. At the iteration termination, the model MAE and RSME are, respectively, 0.04 and 0.03 mm for the training and testing sets in the x-direction of DC1, which are significantly lower than the standard deviation of 0.96 and 1.06 mm from Table 1, indicating a well-trained MLA-TCN model. Additionally, the difference between the MAE and RSME is 0.001 and 0.01 mm for the training and testing sets in the x-direction of DC1, respectively, suggesting a relatively low variance of errors in the training and testing sets.

4.5. Model Performance. The MLA-TCN model's prediction performance is evaluated using 4749 measured samples from January 1, 2008, to December 31, 2020. Figure 10 displays the model's prediction performance at the DC1 station, with 156 sample points selected from the first day of each month during the period. The black line represents the measured displacement value, and the red line indicates the predicted value. The MLA-TCN model accurately predicts the irregular, periodic variation in the dam's displacement, and the predicted values closely match the actual values. Table 2 presents the evaluation metrics at four measurement points. The proposed model's MAE and RSME values are lower than the standard deviation, indicating high accuracy. The coefficients of determination are close to 1.0, suggesting that the model fits the experimental data well. The difference between the MAE and RSME is lower than 0.3 mm, indicating a narrow variance of prediction error and high prediction accuracy. The above results demonstrate that the MLA-TCN model performs excellently in predicting dam displacement.

To simplify, the proposed MLA-TCN model is a multi-output regression analysis model that predicts the displacement of a dam in both x- and y-directions. However, to analyze the model's generalization, four output branches are utilized to predict the displacement at four measurement points. Table 3 shows that the model's performance remains excellent, indicating high generalization. It should be noted that this study only considers two output branches for regression analysis, except for exceptional explanation.

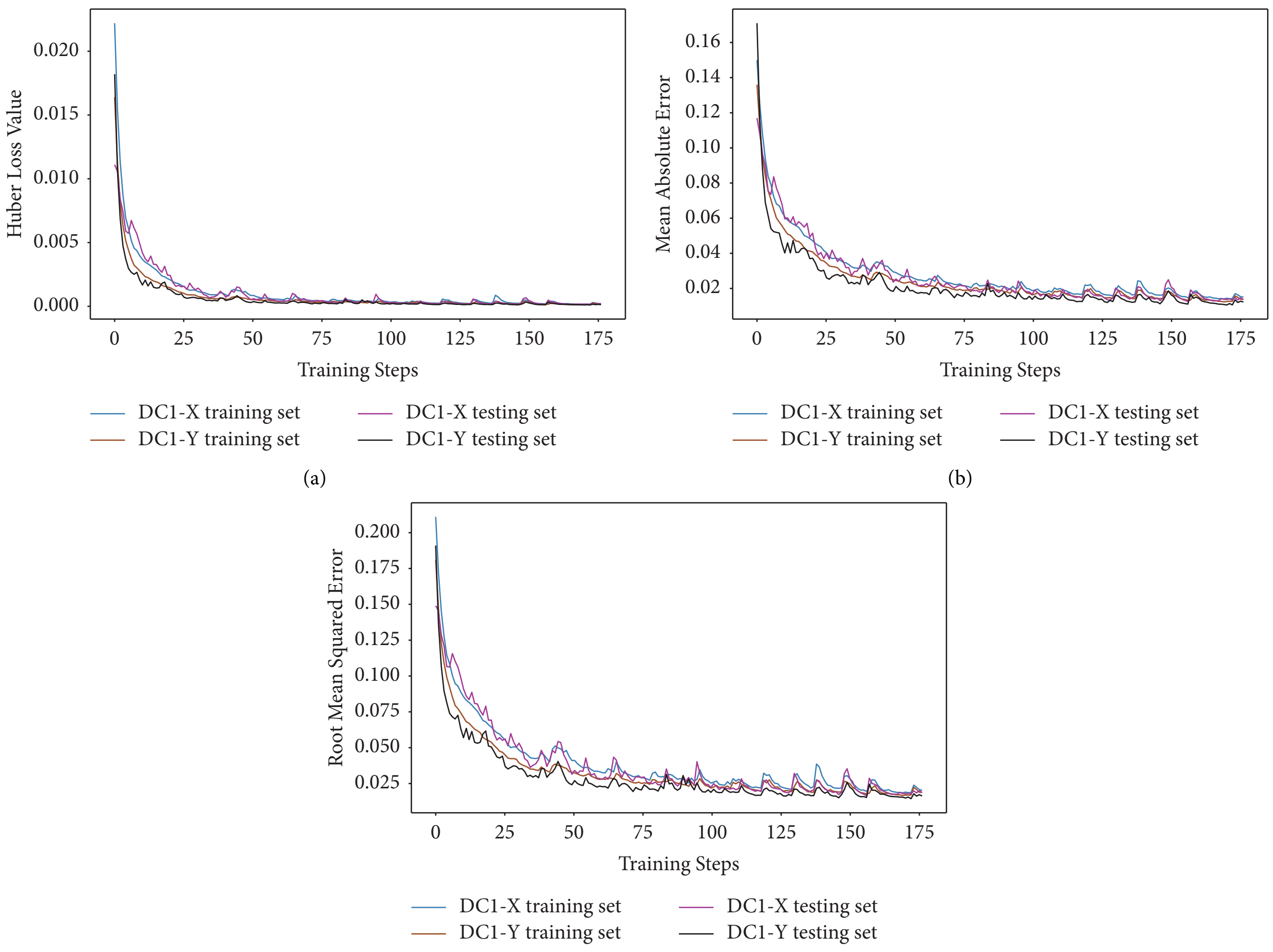
4.6. Analysis of Temporal Sequence Length. This section aims to explore the impact of sequence length on the prediction performance of the MLA-TCN model, a critical aspect in temporal regression analysis for dam displacement prediction. Dam displacement evolution encompasses various temporal scales, including short-term fluctuations and long-term trends, necessitating different temporal intervals for consideration. To investigate this, the model is trained and tested with 15 sequence lengths ranging from 5 to 700 (5, 10, 15, 20, 50, 100, 150, 200, 250, 300, 350, 400, 500, 600, 700), covering both short and long-term processes. The goal of this exploration is to determine the optimal sequence length for accurate dam displacement prediction.

Figure 11 shows the model's performance for each measurement point, with MAE and R^2 values presented for different sequence lengths. The assessment metrics for each measurement point are normalized to a range of 0 to 1.0 and plotted in a scatterplot form, with the horizontal coordinate indicating the temporal sequence length. The results indicate that the MLA-TCN model performs better with longer sequence lengths, but there is no strict physical regulation governing this relationship. Based on this trend, the proposed model selects an optimal temporal sequence length of 100 for its short running time and strong information extraction capability.

Figure 11(a) plots the MAE values for the MLA-TCN model with different sequence lengths. It is evident that the longer the sequence length taken, the lower the prediction error. A power function with formula $y = 1.67x^{-0.648}$ also fits the variation of the mean absolute error with the sequence length.

Figure 11(b) presents the variation of the coefficient of determination of the MLA-TCN model with the sequence length: the longer the sequence length, the higher the regression performance. Furthermore, a power function with the formula $y = 1.67x^{-0.0525}$ can model the variation of R^2 with the sequence length.

The abovementioned findings prove that the previous argument that the prediction performance of the model is relevant to the sequence length is proper. Generally, the performance of the proposed model improves with the length of the temporal sequence. When the sequence length of the MLA-TCN model is taken as 100, its performance is optimal, and further raising the time series length does not enhance its prediction accuracy much. Instead, it may increase the complexity of the model operation, thereby raising the computation costs. This demonstrates that although the proposed model is able to learn a longer-term pattern in the temporal dependencies, the model obtains features from a sequence length of 100 at most; hence, it still has the potential to improve.

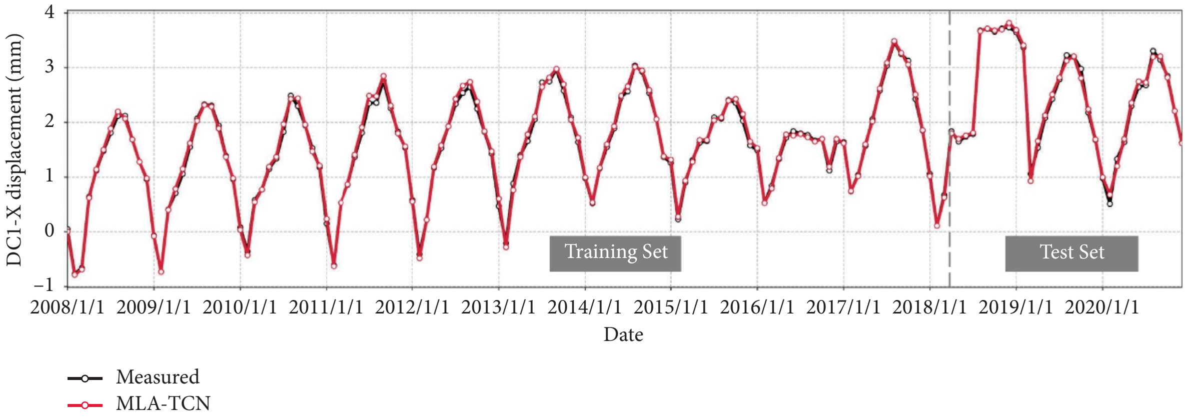


(a)

(b)

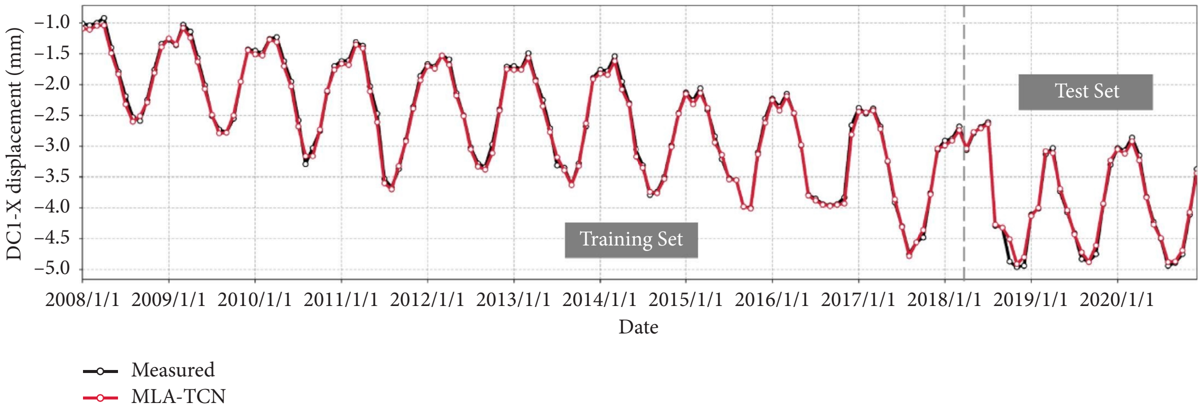
(c)

Figure 9: The errors in the training and testing sets versus the training epoch: (a) the Huber loss; (b) the mean absolute error; (c) the root-mean-square error.



(a)

Figure 10: Continued.



(b)

Figure 10: The performance of the MLA-TCN model: the (a) DC1-X and (b) DC1-Y displacement of the dam.

Table 2: The performance of the MLA-TCN model with two branches.

Observation point	MAE (mm)	RSME (mm)	RSME–MAE (mm)	R^2
DC1-*X*	0.0547	0.0743	0.0196	0.9943
DC1-*Y*	0.0491	0.0635	0.0144	0.9960
DC5-*X*	0.0327	0.0486	0.0159	0.9908
DC5-*Y*	0.0217	0.0311	0.0094	0.9972
DC6-*X*	0.1979	0.2785	0.0806	0.9844
DC6-*Y*	0.1881	0.2648	0.0767	0.9872
ZC2-*X*	0.5578	0.8763	0.3185	0.9500
ZC2-*Y*	0.5528	0.8899	0.3371	0.9680

Table 3: The performance of the MLA-TCN model with four branches.

Observation point	MAE (mm)	RSME (mm)	RSME–MAE (mm)	R^2
DC1-*X*	0.0599	0.0869	0.0270	0.9916
DC1-*Y*	0.0471	0.0635	0.0164	0.9963
DC5-*X*	0.0342	0.0517	0.0175	0.9899
DC5-*Y*	0.0245	0.0343	0.0098	0.9966
DC6-*X*	0.1973	0.2849	0.0876	0.9832
DC6-*Y*	0.1786	0.2609	0.0823	0.9878
ZC2-*X*	0.4616	0.7708	0.3092	0.9648
ZC2-*Y*	0.4934	0.7774	0.2840	0.9753

4.7. Visualization of Weights. In the proposed model, the attention mechanism layer assigns weights according to the importance of the input variables. This section visualizes and analyzes the weight assignment results, presented in the form of three-dimensional waterfall graphs in Figures 12 and 13, to further understand the relationship between the displacement of the dam and each variable.

Figure 12 displays the initial weight allocation of the MLA-TCN model at four measurement points (DC1, DC5, DC6, and ZC2). The *x*-coordinate represents the sequence length, set at 100, the *y*-coordinate signifies the four input variables, and the *z*-coordinate corresponds to the weight. It can be observed that the initial weight values vary across factors, suggesting that the MLA-TCN model can selectively extract valuable information based on the importance of a factor's impact on displacement. The weights of each influencing variable differ across diverse sequence lengths at each measurement point, suggesting that the impact of environmental variables on dam displacement is a dynamic process, rapidly changing with space.

Figure 13 visualizes the influence of four environmental variables (temperature, upstream water level, downstream water level, and time) on dam displacement over time, represented by their weights. It indicates that displacement is primarily influenced by upstream water level, temperature, and time, in alignment with traditional statistical models [36, 37]. This underscores that the MLA-TCN model is capable of deriving a reasonable relationship between displacement and influencing variables through the visualization of the attention mechanism. Moreover, the figure reveals the evolution of displacement allocation over time, indicating that the influence of environmental variables on displacement tends to stabilize over time. For instance, measurement points DC1, DC5, and DC6 are

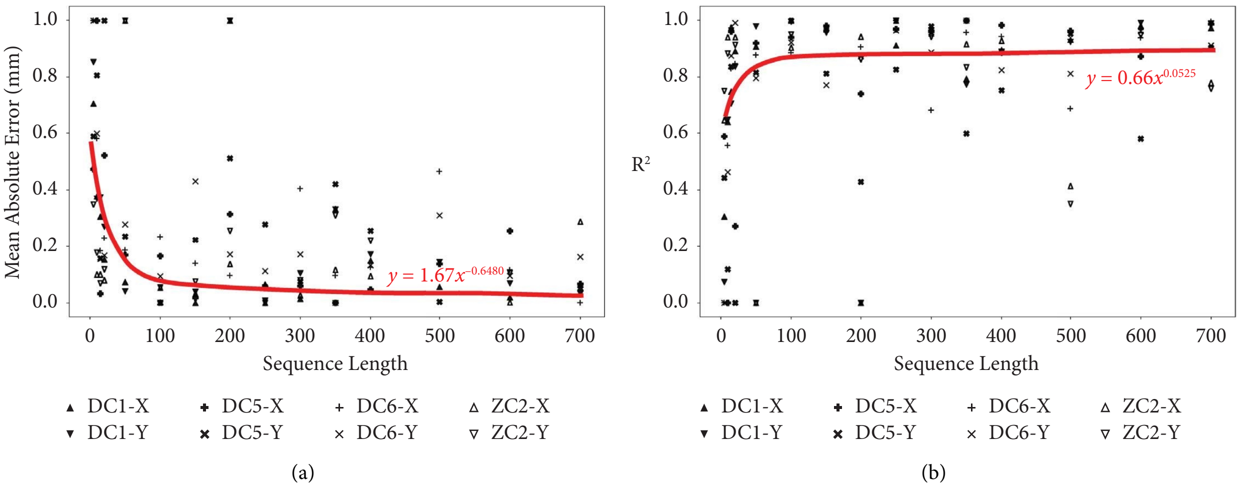


FIGURE 11: The effect of the sequence length on the model performance: (a) the mean absolute error; (b) the coefficient of determination.

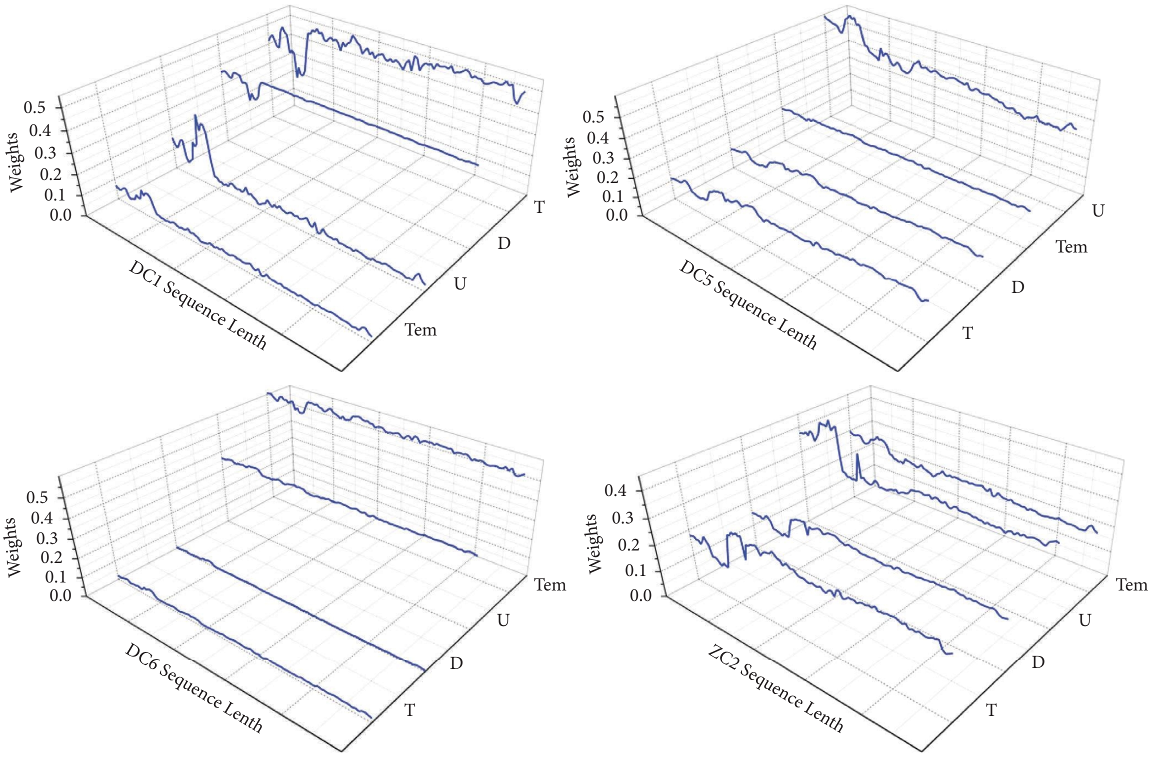


FIGURE 12: The initial visualization of the weights.

primarily affected by time in the later stages, as they are situated near the dam's base and foundation, and their displacements are mostly influenced by physical characteristics such as material aging and structural damage. Conversely, the ZC2 measurement point, located in the middle of the dam crest, is chiefly affected by temperature and upstream water level, as its displacement is predominantly influenced by loads such as water and temperature. This demonstrates that the MLA-TCN model can gain a reasonable understanding of how environmental variables influence displacement over time, as made visible by the attention mechanism.

In conclusion, through weight visualization, the proposed model can not only predict dam displacement but also output the reasonable relationship between dam displacement and each influencing factor, providing an auxiliary tool for understanding the evolution mechanism of dam displacement.

5. Comparing Developed MLA-TCN Model with Other Machine-Learning Methods

This section compares the developed MLA-TCN model with six state-of-the-art machine-learning regression algorithms, namely, the support vector regression (SVR), random forest

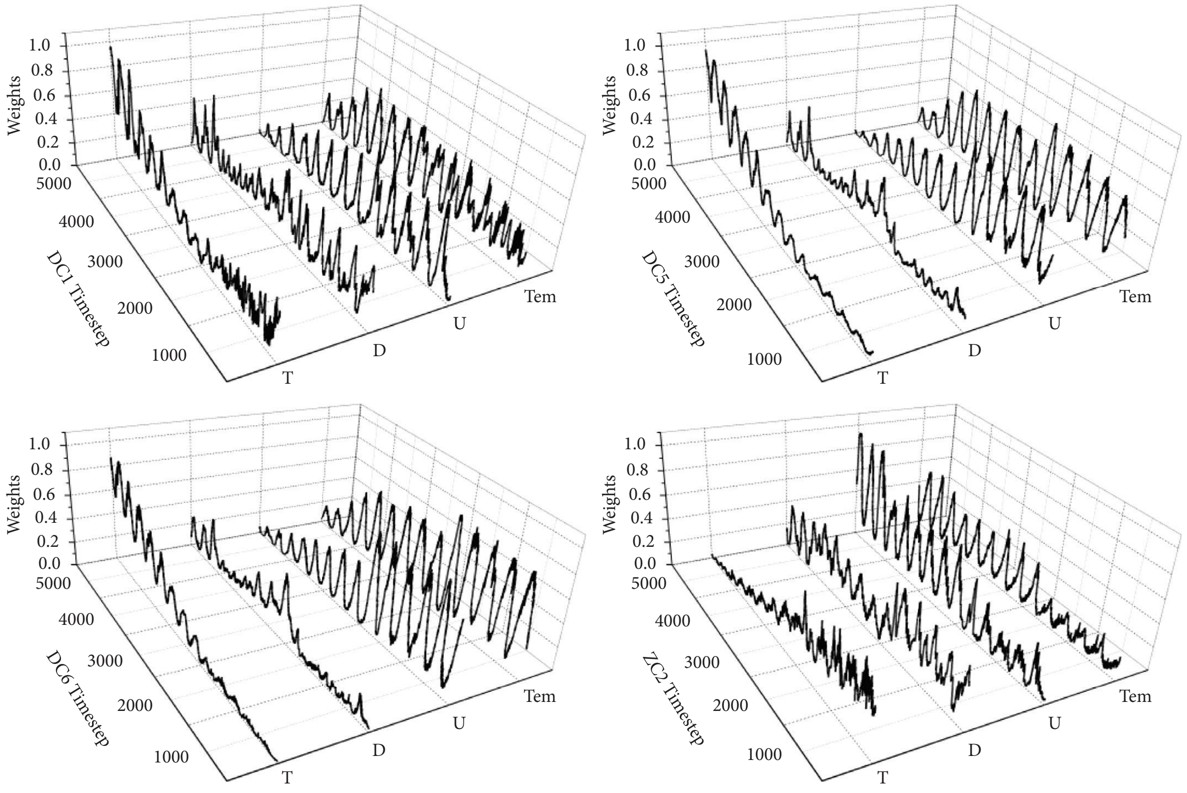


Figure 13: The evolution of the weights.

(RF), LSTM, GRU, long short-term memory network with attention mechanism (ATTLSTM), and TCN, to further demonstrate its superiority. The SVR and RF are the widely applied machine-learning models, while the LSTM, GRU, ATTLSTM, and TCN are representative deep-learning models; the TCN is a temporal convolutional deep-learning model not deployed to the field of dam displacement prediction. These models cover diverse aspects and can provide a comprehensive comparison.

5.1. Parameter Configuration. Before employing the six models to predict the displacement of the dam, we should determine the combination of the hyperparameters affecting the prediction performance of each model so that each model can obtain the optimal performance capability. Therefore, this section adopts the hyperband algorithm to optimize the hyperparameters of the deep-learning models, while setting the number of iteration epochs at 30 and the batch size at 32. The grid-search algorithm is also introduced to optimize the hyperparameters of machine-learning models SVR and RF. Table 4 presents the hyperparameter settings and the optimum results of each model.

5.2. Comparing Performance of Models. The prediction performance of the proposed model and other comparative models is assessed with 4749 samples measured from January 1, 2008, to December 31, 2020. Table 5 presents the evaluation results, including MAE, RMSE, discrepancy, and R^2, at each measurement point. The MAE and RSME of all the models at each measurement point are significantly lower than the standard deviation of the original data. The R^2 values of most models generally reach 0.6, although there are exceptions with some RF and SVR models at certain measurement points falling below this threshold. These results suggest that these models can provide good predictive results in the majority of instances.

For a more intuitive comparison, Figure 14 presents the relative values of evaluation metrics for benchmark models and the proposed MLA-TCN model across different measurement points. The vertical axis represents the ratio of the evaluation metrics between each model and the MLA-TCN model, while the horizontal axis signifies the values of the evaluation metrics for each model. It is noteworthy that MLA-TCN, TCN, ATTLSTM, LSTM, and GRU models are deep-learning models, while RF and SVR models are machine-learning models.

Figure 14 underscores that deep-learning models outperform machine-learning models, substantiating the claim made in the previous introduction. This superior performance of deep-learning models can be credited to their ability to capture temporal sequence information, making them more adept at predicting dam displacement than their machine-learning counterparts. Additionally, the proposed MLA-TCN model demonstrates a superior performance relative to other deep-learning models.

As noted earlier, data at measurement points DC6 and ZC2 exhibit relatively high variability. Among the deep-learning models, the TCN model performs significantly

Table 4: The hyperparameter configuration of the comparison models.

Models	Hyperparameter settings	Optimum results
SVR	$C \in [1, 10]$, $E \in [0.001, 0.15]$	$C = 9.5$ and $E = 0.0945$
RF	$N \in [1, 200]$, $D \in [1, 5]$	$N = 154$ and $D = 4$
LSTM	$U = [16,\ 32,\ 64,\ 128]$, LR = $[0.01, 0.001, 0.0001]$, SL = $[10, 15, 100, 200, 400]$	$U = 64$, LR = 0.001, and SL = 10
GRU	$U = [16,\ 32,\ 64,\ 128]$, LR = $[0.01, 0.001, 0.0001]$, SL = $[10, 15, 100, 200, 400]$	$U = 32$, $LR = 0.01$, and $SL = 10$
ATTLSTM	$U = [16,\ 32,\ 64,\ 128]$, LR = $[0.01, 0.001, 0.0001]$, SL = $[10, 15, 100, 200, 400]$	$U = 64$, $LR = 0.001$, and $SL = 10$
TCN	$F = [1, 16, 32, 64, 128, 256]$, $S \in [1, 9]$, LR = $[0.01, 0.001, 0.0001]$, SL = $[10, 15, 100, 200, 400]$	$F = 32$, $S = 3$, LR = 0.01, and SL = 100

Table 5: The performance of the seven machine-learning models.

		MLA-TCN	TCN	ATTLSTM	LSTM	GRU	RF	SVR
DC1-X	MAE	0.0599	0.1828	0.2660	0.2626	0.3716	0.4183	0.3925
	RSME	0.0869	0.2618	0.3804	0.4103	0.5284	0.5357	0.5080
	RSME–MAE	0.0270	0.0790	0.1144	0.1477	0.1567	0.1174	0.1156
	R^2	0.9916	0.8920	0.8428	0.7764	0.7086	0.5150	0.6109
DC1-Y	MAE	0.0471	0.1551	0.1600	0.1504	0.1913	0.3012	0.3027
	RSME	0.0635	0.2248	0.2611	0.2153	0.2613	0.3874	0.3895
	RSME–MAE	0.0164	0.0698	0.1011	0.0649	0.0700	0.0862	0.0867
	R^2	0.9963	0.9411	0.9321	0.9552	0.9102	0.8227	0.8417
DC5-X	MAE	0.0342	0.0810	0.0941	0.0859	0.1370	0.1705	0.1912
	RSME	0.0517	0.1191	0.1659	0.1515	0.2366	0.2432	0.2581
	RSME–MAE	0.0175	0.0381	0.0717	0.0656	0.0996	0.0727	0.0669
	R^2	0.9899	0.9438	0.8801	0.9033	0.7977	0.6761	0.6518
DC5-Y	MAE	0.0245	0.0601	0.0645	0.0636	0.0716	0.0760	0.1132
	RSME	0.0343	0.0821	0.0916	0.0904	0.0983	0.0973	0.1380
	RSME–MAE	0.0098	0.0220	0.0271	0.0268	0.0267	0.0213	0.0247
	R^2	0.9966	0.9788	0.9760	0.9742	0.9714	0.9715	0.9410
DC6-X	MAE	0.1973	0.3304	0.4604	0.4282	0.5170	0.8435	0.8685
	RSME	0.2849	0.4732	0.7183	0.6156	0.7177	1.0047	1.0694
	RSME–MAE	0.0876	0.1427	0.2579	0.1874	0.2007	0.1612	0.2009
	R^2	0.9832	0.9506	0.8748	0.9268	0.8859	0.7320	0.6850
DC6-Y	MAE	0.1786	0.5080	0.3133	0.3368	0.7540	0.4058	0.7067
	RSME	0.2609	0.7664	0.4477	0.4596	1.4834	0.5361	0.9050
	RSME–MAE	0.0823	0.2583	0.1343	0.1228	0.7294	0.1303	0.1984
	R^2	0.9878	0.8613	0.9605	0.9603	0.9136	0.9435	0.8320
ZC2-X	MAE	0.4616	1.2854	1.0552	1.0567	1.4052	1.6161	2.0032
	RSME	0.7708	1.8661	1.6005	1.6893	2.0675	2.1858	2.5707
	RSME–MAE	0.3092	0.5807	0.5453	0.6326	0.6623	0.5697	0.5675
	R^2	0.9648	0.7694	0.8360	0.7860	0.6244	0.5596	0.4471
ZC2-Y	MAE	0.4934	1.7897	0.9687	0.8011	1.1125	1.8785	2.3633
	RSME	0.7774	2.8954	1.5186	1.2252	1.6342	2.4795	3.1281
	RSME–MAE	0.2840	1.1057	0.5499	0.4241	0.5218	0.6010	0.7648
	R^2	0.9753	0.6153	0.9014	0.9390	0.8677	0.6564	0.3226

better than other comparative models at measurement points DC1 and DC5, while its performance at measurement points DC6 and ZC2 is comparatively poorer. This divergence can be attributed to the TCN model's reliance on dilated convolutional layers for capturing long-term dependencies, as opposed to an explicit memory mechanism, making it more sensitive to data quality. The MLA-TCN model, however, circumvents this limitation by enhancing the model's information extraction capacity through the attention mechanism, thereby compensating for the TCN model's dependence on data quality. Hence, the MLA-TCN model exhibits superior performance in predicting dam displacement.

While the ATTLSTM model incorporates an attention mechanism, Figure 14 demonstrates that the combination of attention and LSTM is not as effective as the combination with TCN. As further substantiated by Table 5, the ATTLSTM model displays unstable prediction errors, characterized by a relatively large MAE and a smaller RSME, leading to a significant discrepancy between the two metrics. This suggests a high variance in prediction errors and a lower prediction accuracy. The root cause of this lies in LSTM's relatively lesser proficiency at capturing long-term dependencies compared to TCN. In contrast, the combination of TCN and attention is adept at capturing both local and global dependencies, thereby

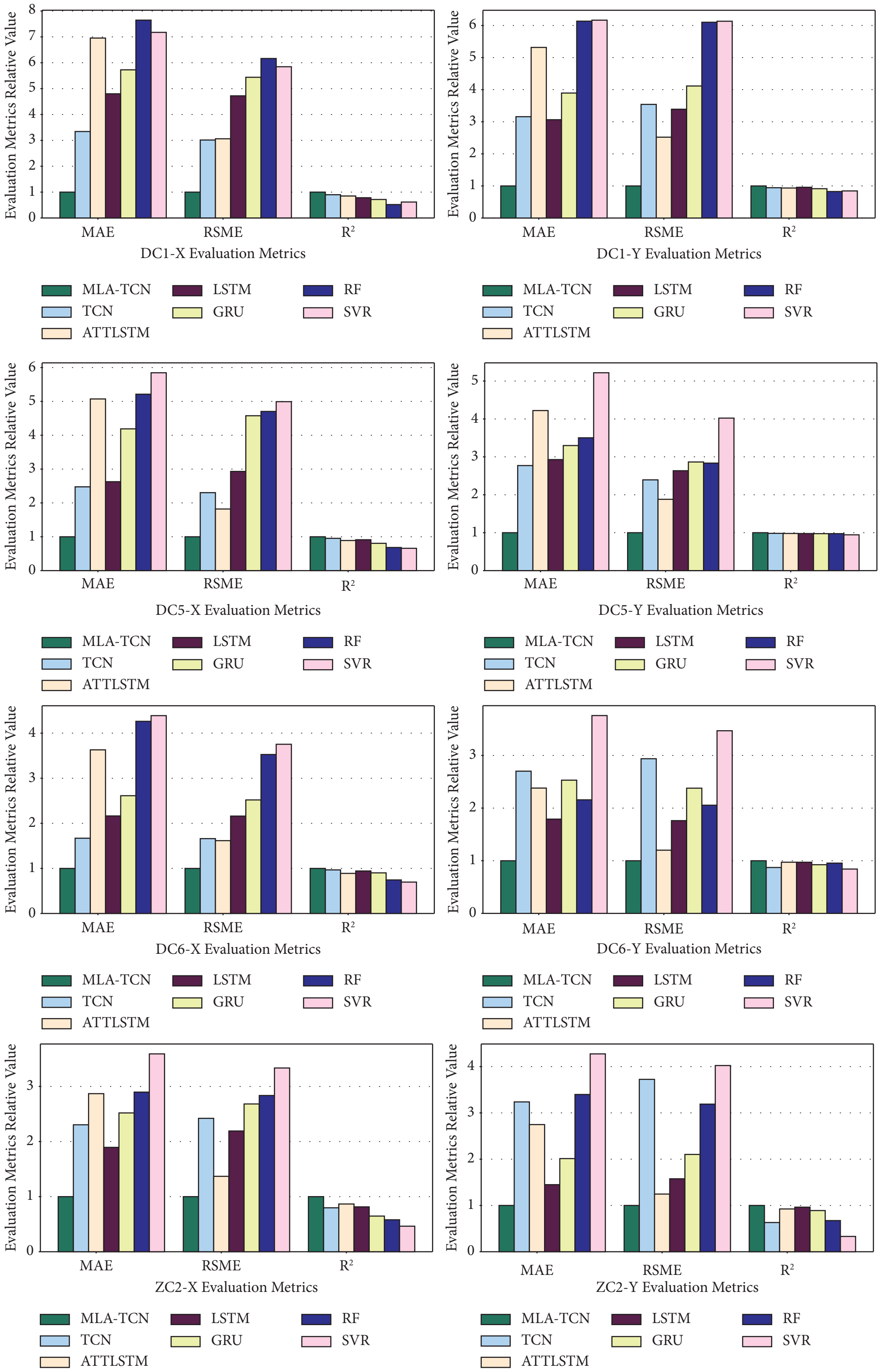


Figure 14: The performance of the established model compared with seven machine-learning models.

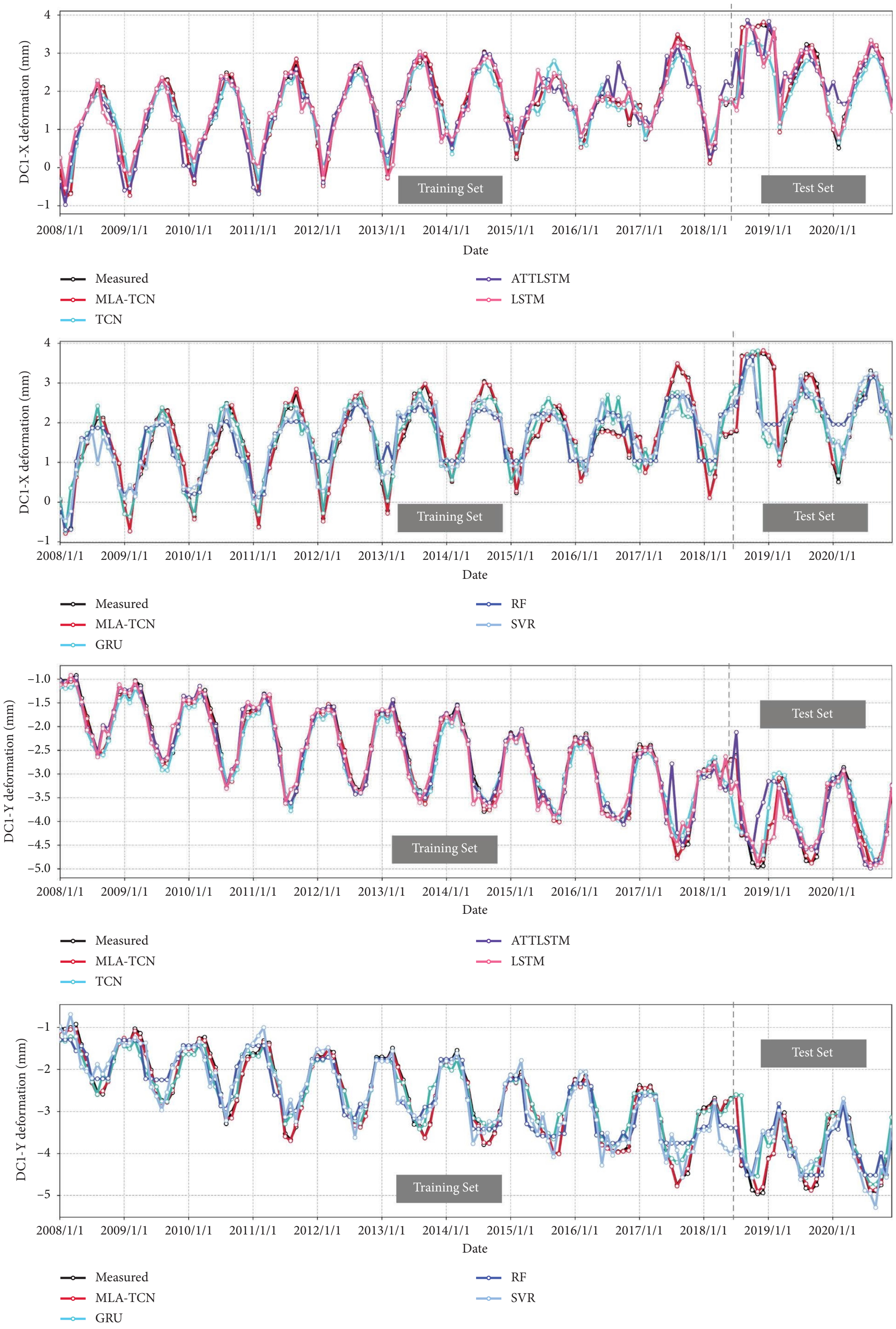


Figure 15: Comparing the results predicted by the MLA-TCN model and the seven machine-learning models with the measured data.

facilitating a more comprehensive understanding of the temporal dynamics intrinsic to the data. Consequently, the MLA-TCN model proves to be more suitable for predicting dam displacement.

Figure 15 compares the prediction results of all the models with the measured data at measurement point DC1 to explicitly observe the prediction performance of the established model and seven machine-learning models. The *y*-axis represents the displacement of the dam, and the x-axis indicates the date; the red curve denotes the prediction of the MLA-TCN model, and the results predicted by the other models are plotted in different colors. Figure 15 illustrates that all models exhibit relatively accurate predictions of the periods of dam displacement changes and the peak-to-peak values within the training set. However, a noticeable distinction emerges in the testing set, where the MLA-TCN model outperforms the others, highlighting its superior prediction performance and robust generalization capability.

6. Conclusions

This work developed a multioutput model for predicting the displacement of concrete dams based on the temporal convolutional network with the attention mechanism. The model introduced the attention mechanism and multioutput regression branch based on the temporal convolutional network and could effectively estimate the dam displacement. The applicability of the proposed model was confirmed by a total of 5104 samples concerning a concrete gravity dam from January 1, 2007, to December 31, 2020. Comparing the established model with seven state-of-the-art machine-learning models confirmed its superior performance. From the abovementioned findings, the following main conclusions could be drawn:

(1) The MLA-TCN model ultimately adopted the measured data as its input, which could extract the practical features without sophisticated preprocessing and predict the dam displacement realistically and accurately.

(2) Multioutput branches provided the MLA-TCN model with a high generalization capability to simultaneously predict the displacement of multiple measurement points. In addition, expanding the output branches did not affect the prediction performance of the model.

(3) The prediction performance of the model correlated with the sequence length: the performance capability of the proposed model generally increased with the length of the temporal sequence. The TCN enabled the MLA-TCN model to learn long-term patterns from long sequences, and the model performed optimally at a sequence length of 100.

(4) The attention mechanism not only contributed to the powerful feature-screening capability of the MLA-TCN model but also could increase the interpretability of the model by visualizing the weights. The evolution of the dam displacement was complex, and the effect of environmental variables on it changed with time and space. The MLA-TCN model could provide the physical interpretation for the evolution of the dam displacement by visualizing the weights.

(5) The MLA-TCN model outperformed seven state-of-the-art machine-learning models: the ATTLSTM, TCN, LSTM, GRU, RF, and SVR. Therefore, the developed model could provide an effective method to estimate the displacement of concrete dams.

The developed model does have its limitations. First, the quality of monitored data dramatically impacts the performance of the MLA-TCN model, which makes it challenging to diagnose anomalies and recover them autonomously, thereby limiting the practical application of the model. Moreover, the spatial generalization capability of the model, such as the simultaneous spatial and temporal prediction of the displacement of different dams, remains to be enhanced. Therefore, we will continue our research in the future by considering the above concerns.

Data Availability

The data supporting the current study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

We would like to thank Zhiwei Liao for guidance in the deep-learning algorithm. This research was supported by the National Science Foundation (grant no. 51979244) and the Zhejiang Provincial Natural Science Foundation of China for Young Scholars (grant no. LQ20A020009).

References

[1] M. Li, M. Li, Q. Ren, H. Li, and L. Song, "DRLSTM: a dual-stage deep learning approach driven by raw monitoring data for dam displacement prediction," *Advanced Engineering Informatics*, vol. 51, Article ID 101510, 2022.

[2] B. Li, J. Yang, and D. Hu, "Dam monitoring data analysis methods: a literature review," *Structural Control and Health Monitoring*, vol. 27, no. 3, 2020.

[3] F. Kang, J. Liu, J. Li, and S. Li, "Concrete dam deformation prediction model for health monitoring based on extreme learning machine," *Structural Control and Health Monitoring*, vol. 24, no. 10, Article ID e1997, 11 pages, 2017.

[4] J. Mata, "Interpretation of concrete dam behaviour with artificial neural network and multiple linear regression models," *Engineering Structures*, vol. 33, no. 3, pp. 903–910, 2011.

[5] H. Su, Z. Chen, and Z. Wen, "Performance improvement method of support vector machine-based model monitoring dam safety: performance Improvement Method of Monitoring Model of Dam Safety," *Structural Control and Health Monitoring*, vol. 23, no. 2, pp. 252–266, 2016.

[6] B. Wei, D. Yuan, H. Li, and Z. Xu, "Combination forecast model for concrete dam displacement considering residual correction," *Structural Health Monitoring*, vol. 18, no. 1, pp. 232–244, 2019.

[7] X. Li, Z. Wen, and H. Su, "An approach using random forest intelligent algorithm to construct a monitoring model for dam safety," *Engineering with Computers*, vol. 37, no. 1, pp. 39–56, 2021.

[8] D. Yuan, C. Gu, X. Qin, C. Shao, and J. He, "Performance-improved TSVR-based DHM model of super high arch dams

using measured air temperature," *Engineering Structures*, vol. 250, Article ID 113400, 2022.

[9] Q. Ren, M. Li, S. Bai, and Y. Shen, "A multiple-point monitoring model for concrete dam displacements based on correlated multiple-output support vector regression," *Structural Health Monitoring*, vol. 21, no. 6, pp. 2768–2785, 2022.

[10] H. Su, Z. Chen, and Z. Wen, "Performance improvement method of support vector machine-based model monitoring dam safety," *Structural Control and Health Monitoring*, vol. 23, pp. 252–266, 2016.

[11] B. Dai, C. Gu, E. Zhao, and X. Qin, "Statistical model optimized random forest regression model for concrete dam deformation monitoring," *Structural Control and Health Monitoring*, vol. 25, no. 6, 2018.

[12] F. Salazar, M. Á. Toledo, E. Oñate, and B. Suárez, "Interpretation of dam deformation and leakage with boosted regression trees," *Engineering Structures*, vol. 119, pp. 230–251, 2016.

[13] C. Y. Kao and C. H. Loh, "Monitoring of long-term static deformation data of Fei-Tsui arch dam using artificial neural network-based approaches: long-term static deformation data of fei-tsui arch dam," *Structural Control and Health Monitoring*, vol. 20, no. 3, pp. 282–303, 2013.

[14] Q. Ren, M. Li, L. Song, and H. Liu, "An optimized combination prediction model for concrete dam deformation considering quantitative evaluation and hysteresis correction," *Advanced Engineering Informatics*, vol. 46, Article ID 101154, 2020.

[15] F. Salazar, R. Morán, M. Toledo, and E. Oñate, "Data-based models for the prediction of dam behaviour: a review and some methodological considerations," *Archives of Computational Methods in Engineering*, vol. 24, no. 1, pp. 1–21, 2017.

[16] W. Liu, J. Pan, Y. Ren, Z. Wu, and J. Wang, "Coupling prediction model for long-term displacements of arch dams based on long short-term memory network," *Structural Control and Health Monitoring*, vol. 27, no. 7, 2020.

[17] X. Qu, J. Yang, and M. Chang, "A deep learning model for concrete dam deformation prediction based on RS-LSTM," *Journal of Sensors*, vol. 2019, Article ID 4581672, 14 pages, 2019.

[18] X. Shu, T. Bao, Y. Li, J. Gong, and K. Zhang, "Vae-Talstm: A temporal attention and variational autoencoder-based long short-term memory framework for dam displacement prediction," *Engineering with Computers*, vol. 38, 2021.

[19] K. T. T. Bui, J. F. Torres, D. Gutiérrez-Avilés, V. H. Nhu, D. T. Bui, and F. Martínez-Álvarez, "Deformation forecasting of a hydropower dam by hybridizing a long short-term memory deep learning network with the coronavirus optimization algorithm," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 11, pp. 1368–1386, 2022.

[20] Z. Wen, R. Zhou, and H. Su, "MR and stacked GRUs neural network combined model and its application for deformation prediction of concrete dam," *Expert Systems with Applications*, vol. 201, Article ID 117272, 2022.

[21] B. Yang, T. Xiao, L. Wang, and W. Huang, "Using complementary ensemble empirical mode decomposition and gated recurrent unit to predict landslide displacements in dam reservoir," *Sensors*, vol. 22, no. 4, p. 1320, 2022.

[22] Y. Li, T. Bao, Z. Gao et al., "A new dam structural response estimation paradigm powered by deep learning and transfer learning techniques," *Structural Health Monitoring*, vol. 21, no. 3, pp. 770–787, 2021.

[23] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, http://arxiv.org/abs/1803.01271.

[24] Y. Cao, Y. Ding, M. Jia, and R. Tian, "A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings," *Reliability Engineering & System Safety*, vol. 215, Article ID 107813, 2021.

[25] Y. Wang, J. Chen, X. Chen et al., "Short-term load forecasting for industrial customers based on TCN-LightGBM," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1984–1997, 2021.

[26] Z. Gan, C. Li, J. Zhou, and G. Tang, "Temporal convolutional networks interval prediction model for wind speed forecasting," *Electric Power Systems Research*, vol. 191, Article ID 106865, 2021.

[27] A. Veltman, D. W. J. Pulle, and R. W. De Doncker, "The transformer," *Power Systems*, pp. 47–82, Springer, Berlin, Germany, 2016.

[28] D. Yang, C. Gu, Y. Zhu et al., "A concrete dam deformation prediction method based on lstm with attention mechanism," *IEEE Access*, vol. 8, pp. 185177–185186, 2020.

[29] Q. Ren, M. Li, H. Li, and Y. Shen, "A novel deep learning prediction model for concrete dam displacements using interpretable mixed attention mechanism," *Advanced Engineering Informatics*, vol. 50, Article ID 101407, 2021.

[30] S. Chen, C. Gu, C. Lin C, and M. A. Hariri-Ardebili, "Prediction of arch dam deformation via correlated multi-target stacking," *Applied Mathematical Modelling*, vol. 91, pp. 1175–1193, 2021.

[31] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, September 2015.

[32] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of the 4th Int. Conf. Learn. Represent. ICLR 2016-Conf. Track Proc.*, San Juan, Puerto Rico, USA, May 2016.

[33] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 45, pp. 770–778, 2016.

[34] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[35] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: bandit-based configuration evaluation for hyperparameter optimization," in *Proceedings of the 5th Int. Conf. Learn. Represent. ICLR 2017-Conf. Track Proc.*, pp. 1–15, Toulon, France, April 2017.

[36] P. Léger and M. Leclerc, "Hydrostatic, temperature, time-displacement model for concrete dams," *Journal of Engineering Mechanics*, vol. 133, 2007.

[37] Y. Shi, J. Yang, J. Wu, and J. He, "A statistical model of deformation during the construction of a concrete face rockfill dam," *Structural Control and Health Monitoring*, vol. 25, 2018.