machine learning & knowledge extraction

MDPI

Article

Fine-Tuned Nonlinear Autoregressive Recurrent Neural Network Model for Dam Displacement Time Series Prediction

Vukašin Ćirović [1], Vesna Ranković [2,*], Nikola Milivojević [1], Vladimir Milivojević [1] and Brankica Majkić-Dursun [1]

[1] Jaroslav Černi Water Institute, 11000 Belgrade, Serbia; vukasin.cirovic@jcerni.rs (V.Ć.); nikola.milivojevic@jcerni.rs (N.M.); vladimir.milivojevic@jcerni.rs (V.M.); brankica.majkic-dursun@jcerni.rs (B.M.-D.)
[2] Department for Applied Mechanics and Automatic Control, Faculty of Engineering, University of Kragujevac, 34000 Kragujevac, Serbia
* Correspondence: vesnar@kg.ac.rs

Abstract

Dam monitoring data are nonlinear and nonstationary time series. Most existing data-driven dam displacement models are developed independently for each measuring point, disregarding the fact that a dam is a complex structure composed of various interconnected elements that form a unified whole. Regardless of the dam type, all points on the dam are exposed to the same external environmental influences. To account for the correlation between displacement time series at different points, this paper proposes a novel fine-tuned deep-learning nonlinear autoregressive (NAR) model based on a Long Short-Term Memory (LSTM) network for predicting dam tangential displacement, and a new method for generating source data to train the base model. The models for three measuring points were developed and tested on experimental data collected over a period of slightly more than twelve years. Compared with the model without fine-tuning, the proposed approach achieves an average mean square error (MSE) reduction of 80.68% on the training set and 65.79% on the test set, as well as an average mean absolute error (MAE) reduction of 51.05% and 52.62%, respectively. Furthermore, the proposed model outperforms Random Forest (RF), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP) models for dam displacement prediction.

Keywords: NAR-LSTM; transfer learning; STL decomposition; DTW; time series prediction; dam displacement

Academic Editor: Jaroslaw Krzywanski

Received: 21 February 2026
Revised: 31 March 2026
Accepted: 2 April 2026
Published: 5 April 2026

1. Introduction

Dams are inherently associated with a certain level of risk that can pose potential danger to people and downstream infrastructure. This risk can never be completely eliminated. To monitor the structural safety of a dam, a complex technical monitoring system with a significant number of sensors of various types is designed and installed at numerous locations within the dam. One of the crucial steps in the process of continuous dam safety monitoring is determining the consistency between measured values and their corresponding predicted values, obtained through mathematical models. The dam displacement time series is an essential indicator of dam behavior and shows nonstationary and nonlinear features affected by time-varying variables [1]. This structural health indicator depends on temperature and water level, degradation material properties, structural damage, seepage coupling, joint fissures, and other environmental factors [2].

In recent decades, various models have been developed for dam health monitoring that can be classified into deterministic, data-driven, and hybrid models.

The performance of deterministic models based on the finite element method depends on the element type, mesh size discretization, knowledge of the mechanical and thermal properties of the concrete dam and rock foundation, and the choice of boundary conditions. Finite element calculations require model calibration to estimate parameter values and are very time and memory intensive [3,4]. This deterministic method can be applied during the reservoir filling periods [5].

The performance of statistical models and the significant reduction in the false alarm rate largely depend on the quality of the collected data [6]. It is difficult to determine whether abnormal data are caused by dam damage and performance degradation or by sensor errors, sampling errors, or poor data transmission quality [7,8]. The statistical Hydrostatic-Season-Time (HST) model, which is based on multiple linear regression, is widely used for dam displacement prediction because of its simple structure [9,10]. Hybrid models that integrate data-driven techniques with the finite element method can successfully address the limitations of both statistical and deterministic models [11].

The long-term prediction of dam displacement time series is challenging because of their nonlinear and non-stationary characteristics [12]. In the literature, various artificial intelligence (AI) methods have been proposed to improve the modeling of nonlinear and uncertain dam displacement functions. The Support Vector Machine (SVM) method is well suited for accurate modeling of nonlinear, non-stationary processes with uncertainties [13]. Nonlinear autoregressive support vector regression models with exogenous inputs [14], hybrid SVM-based models [15], SVM models optimized to minimize the mean square error of fitting [16], SVM models optimized with the chaotic particle swarm optimization algorithm [17], and SVM-based models combined with other methods [18] have been developed to estimate dam displacement. The multilayer perceptron [19–21], radial basis function neural network [22], and neuro-fuzzy system [23] have also been applied to dam behavior prediction. Salazar et al. [24] compared the performance of models based on five machine learning techniques with those of traditional statistical techniques for predicting displacement and leakage.

In recent years, recurrent long short-term memory neural networks have been introduced to predict dam behavior because of their proven effectiveness in processing time series data [25,26]. Ren et al. [27] constructed a dam displacement model using an encoder-decoder architecture that includes LSTM layers, factorial and temporal attention mechanisms, and a fully connected layer. Wen et al. [28] developed a novel structure based on multiple regression and a stacked gated recurrent unit neural network for predicting the hydrostatic, temperature, aging, and residual components of dam displacement.

A hybrid approach combining LSTM with Kolmogorov–Arnold networks and a dual-stage attention mechanism, including factor and temporal attention, has been proposed by Xu et al. [29]. Bidirectional gated recurrent unit (BiGRU) models with sliding windows have been proposed to capture time-lag effects of environmental factors on dam displacement and improve prediction accuracy [30].

A systematic review of dam health monitoring models, their applications, advantages, limitations, and recommendations for selecting the most suitable model for a given dam was presented by Prakash et al. [31].

Most existing concrete dam displacement models are based on data from single measuring points and typically do not consider the spatiotemporal correlations between monitoring points [32]. The displacements of monitoring points on a dam often have an annual periodicity because they are affected by seasonal variables such as the reservoir water level and temperature. A long-term correlation is present in the dam deformation

time series with cyclical variations [33]. The time series of dam displacements are similar because all the measurement points are subjected to similar environmental factors. Additionally, the spatial correlation between displacement monitoring points is a result of the structural integrity of the dam [34].

AI models for dam displacement prediction incorporating spatial correlation have been developed for a concrete arch dam using convolutional neural networks [35] and for a concrete gravity dam using support vector machines [32].

Deep learning architectures and transfer learning have been applied to investigate spatio-temporal correlations among dam displacement measurement points and to model nonstationary and nonlinear dam deformation time series. Hybrid deep learning structures, such as one-Dimensional Residual network-LSTM [36], have been proposed for dam deformation prediction. The LSTM network is combined with DenseNet, and the obtained results show that the proposed architecture can significantly improve the prediction performance of the dam displacement model [37].

Li et al. [38] applied a deep learning architecture for high arch dam displacement prediction and utilized transfer learning to transfer knowledge from a base model trained with data from a typical monitoring point at the dam crest. The proposed model consists of a 1D convolutional layer, a bidirectional gated recurrent unit with a self-attention mechanism, and a fully connected layer.

Chen et al. [39] proposed a deep model architecture with a one-dimensional convolutional layer, two LSTM layers, and a fully connected layer to model dam displacement with insufficient measurement data. The model inputs are the periodic, trend, and random components of a one-dimensional sequence of dam deformation. The prediction of the displacement for the next day was based on the radial displacement data from the previous days. A dam with sufficient data to train the original model was selected. Then, the parameters of the CNN and LSTM layers of the pre-trained model were frozen, and the target dam displacement data were used to fine-tune the fully connected layer of the target model.

Transfer learning is widely used in deep learning models for time series analysis and to improve prediction accuracy [40]. A review of the literature indicates that deep transfer learning models have been developed and applied for predicting the displacement of concrete arch dams [38–41].

Considering that LSTM is among the most effective methods for analyzing and forecasting time series data [42], this paper introduces a novel NAR-LSTM model with transfer learning for dam displacement prediction for concrete gravity dams. In the fine-tuning phase of the proposed model, the weights of all layers, both shallow and deep, are adjusted with varying learning rates, unlike existing deep transfer learning models for dam behavior, where shallow layers are typically frozen. Shallow layers are assigned smaller learning rates, enabling them to adapt to the target domain while retaining the key features learned from the source domain. One of the contributions of this study is the unique method for generating the source domain. Specifically, a set of potential source time series is generated from the seasonal, trend, and residual components obtained by STL decomposition of the displacements recorded at all the dam monitoring points. The source series most similar to the target data is selected based on the DTW distance. The proposed model is initially trained using the generated time series, thus accounting for the existing correlation among the monitoring points. The pre-trained model is fine-tuned with the target sensor monitoring data. This approach achieves excellent predictive performance while avoiding the unnecessary complexity of deep learning architectures.

The remainder of the article is structured as follows: In Section 2, the flowchart of the proposed method is shown, and the theory for understanding the proposed paradigm

is presented. Basic data on Hydroelectric Power Plant Đerdap 2 and the presentation of data collected from dam monitoring systems are provided in Section 3. The results and corresponding discussion are shown in Sections 4 and 5, respectively. Section 6 contains concluding remarks.

2. Materials and Methods

The flowchart of the proposed deep learning prediction method for dam displacement is shown in Figure 1. The first step in building a deep learning model for dam displacement prediction is data collection from structural health monitoring systems and data preprocessing, which includes outlier removal, missing data imputation, and normalization.

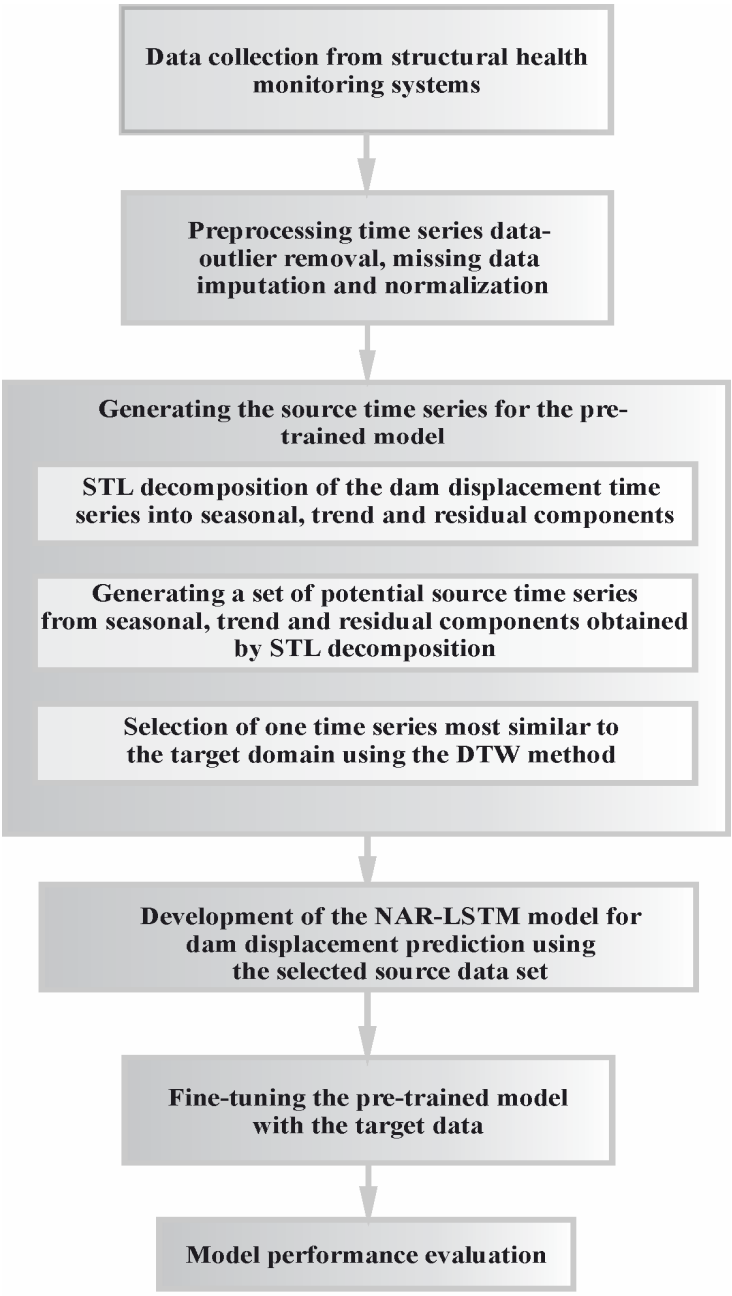

Figure 1. The flowchart of the proposed deep learning prediction method for dam displacement.

In the next step, the STL decomposition of the dam displacement time series is performed, and potential source time series are generated for each monitoring point. The DTW

method is then used to select the time series from the set of generated candidates that is most similar to the target domain. In the following phase, the NAR model based on the LSTM network is trained using the source domain data. Finally, the pre-trained model is fine-tuned using the target domain data.

2.1. STL Decomposition of the Time Series

The dam displacement time series of the ith point at time instant k ($y_i(k)$) is decomposed into seasonal ($S_i(k)$), trend ($T_i(k)$), and residual ($R_i(k)$) items using STL methodology [43]. The STL decomposition can be formulated as:

$$y_i(k) = S_i(k) + T_i(k) + R_i(k), \quad k = 1, N \tag{1}$$

where N represents the number of samples.

The periodic seasonal component of the dam displacement is caused by seasonal temperature fluctuations and changes in water level. The trend component represents long-term changes in the dam displacement time series, primarily related to the aging process, which includes the degradation of material properties. The remainder component is affected by random and uncertain factors.

The STL decomposition algorithm incorporates two recursive procedures: an inner loop and an outer loop. Each iteration of the inner loop includes seasonal and trend smoothing to update both components. The steps within each iteration of the inner loop are as follows: the detrended time series is obtained by subtracting the trend values from the original series; the smoothed cycle-subseries of the detrended series are computed; the low-pass filtering of the smoothed cycle-subseries is performed; the deseasonalized series is calculated by removing the seasonal component from the original time series; and the trend component is obtained by smoothing the deseasonalized series. In the outer loop, the remainder is calculated by subtracting the trend and seasonal components from the original series. Additionally, the outer loop computes the robustness weights that are used during the next execution of the inner loop.

The parameters of the STL algorithm are as follows: the number of samples in each period of the seasonal component ($n_{(p)}$); the number of passes through the inner loop ($n_{(i)}$), and the outer loop ($n_{(o)}$); the smoothing parameters for the low-pass filter ($n_{(l)}$), the trend component ($n_{(t)}$), and the seasonal component ($n_{(s)}$), respectively. In practical applications of STL decomposition, most of the parameters are automatically determined, and the iterative process stops once the convergence criterion is met [43].

2.2. Time Series Similarity Measurement Based on the DTW

Dynamic time warping (DTW) is applied to determine the similarity between two time series [44]. DTW measure between time series $y_i = \begin{Bmatrix} y_i(1) & \dots & y_i(N-1) & y_i(N) \end{Bmatrix}$ and $y_j = \begin{Bmatrix} y_j(1) & \dots & y_j(M-1) & y_j(M) \end{Bmatrix}$ can be written as:

$$DTW(y_i, y_j) = D(N, M) \tag{2}$$

N, M represents the number of data points of the time series y_i and y_j, respectively, $D(N, M)$ denotes the value of the cost matrix $D(n, m)$ at the Nth row and the Mth column.

The cost matrix is determined by the recursive formula as follows:

$$D(n, m) = d(n, m) + \min \begin{cases} D(n-1, m) \\ D(n, m-1) \\ D(n-1, m-1) \end{cases} \tag{3}$$

$d(n,m)$ represents the distance between two points $y_i(n)$ and $y_j(m)$ and can be calculated as:

$$d(n,m) = |y_i(n) - y_j(m)| \quad (4)$$

2.3. Generating the Source Datasets

The selection of the source domain is an important step in the development of a transfer learning model and is usually based on an assessment of the similarity between the target and potential source datasets.

In this paper, a new methodology is proposed for generating the source dataset for the pre-trained displacement model of the ith dam point. The source dataset for the ith point can be composed of the seasonal ($S_j(k)$), trend ($T_l(k)$), and residual ($R_m(k)$) components of the other measuring points. Let the target dataset be the ith point $y_i = \begin{Bmatrix} y_i(1) & y_i(2) & \dots & y_i(N-1) & y_i(N) \end{Bmatrix}$, then the source dataset for the ith point can be generated from the components of the other points as:

$$y_{ii}(k) = S_j(k) + T_l(k) + R_m(k), \quad j,l,m = 1, 2, \dots, Z \text{ and } j,l,m \neq i, \quad k = 1,2,\dots N \quad (5)$$

where $S_j(k)$, $T_l(k)$ and $R_m(k)$ are the seasonal, trend, and residual components at the jth, lth, and mth points, respectively, at the kth time instance; Z is the number of measuring points for which prediction models are formed.

The total number of potential source time series for the ith point is equal to the number of variations with repetition:

$$\overline{V}^3_{(Z-1)\cdot} = [(Z-1)]^3 \quad (6)$$

One source time series is selected from the set of potential source time series y_{ii}, that is most similar to the target domain. This series has the smallest *DTW* distance to y_i.

2.4. The Nonlinear Autoregressive (NAR) Model and Transfer Learning

The NAR model is an important class of nonlinear identification methods that has been shown to be very effective for predicting time series data [45]. The target NAR prediction model for the tangential displacement of the ith dam point can be defined as follows:

$$y_{ip}(k) = f_i(y_i(k-1), y_i(k-2), \dots, y_i(k-n_{io}), \boldsymbol{\varphi}_i) \quad (7)$$

where f_i is the target predictive function of the ith model, $y_i(k-j)$, $j = 1, 2, \dots, n_{io}$ are the measured values of the dam tangential displacement at the ith dam point, $\boldsymbol{\varphi}_i$ denotes the parameter vector of the ith target model, and n_{io} is the number of lags of the measured values (y_i). The NAR basic model for the ith point can be represented as:

$$y_{iip}(k) = f_{ii}(y_{ii}(k-1), y_{ii}(k-2), \dots, y_{ii}(k-n_{io}), \boldsymbol{\varphi}_{ii}) \quad (8)$$

where f_{ii} is the source predictive function, $y_i(k-j)$, $j = 1, 2, \dots, n_{io}$ are the values of the generated source time series at the ith dam point, $\boldsymbol{\varphi}_{ii}$ is the parameter vector of the ith source model.

In this paper, a model consisting of two LSTM layers and a fully connected layer is proposed for approximating the source predictive function f_{ii} and the target predictive function f_i.

First, the source model for the ith dam measurement point is trained using data from the source domain: $D_s = \{((y_{ii}(k-1), y_{ii}(k-2), \dots, y_{ii}(k-n_{io})), y_{ii}(k))\}_{k=n_{io}+1}^{N}$.

The obtained pre-trained model is fine-tuned on the target domain for the ith dam point: $D_T = \{((y_i(k-1), y_i(k-2), \dots, y_i(k-n_{io})), y_i(k))\}_{k=n_{io}+1}^{N}$ in order to determine the target predictive function.

Transfer learning improves the prediction accuracy of the target nonlinear function [46], f_i (Equation (7)) by using knowledge from a pre-trained model. During the transfer learning phase, the layers of the pre-trained model can be completely frozen or assigned a smaller learning rate to preserve the learned features.

2.5. Long-Short Term Memory (LSTM)

Long short-term memory neural networks are a type of recurrent neural network (RNN) developed by Hochreiter et al. [47]. Since its original formulation, several different implementations of the LSTM have been introduced to improve model performance [48]. In this study, LSTM architecture without peephole connections is employed.

The basic unit of a hidden layer is a memory block with memory cells. The LSTM block at time step $k-j$ is shown in Figure 2.

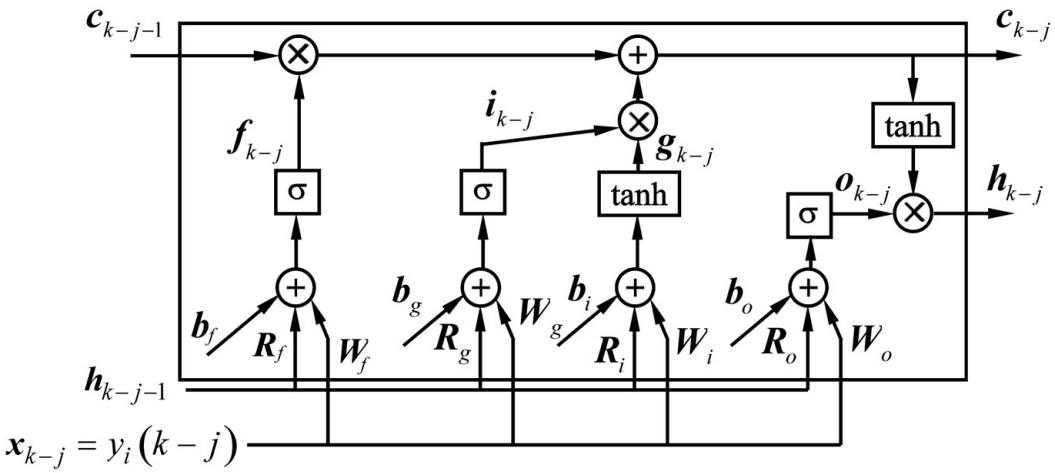

Figure 2. Architecture of an LSTM block at time step $k-j$.

The input, output and current state of the LSTM block at time $k-j$ are denoted as $\boldsymbol{x}_{k-j}$, $\boldsymbol{h}_{k-j}$, $\boldsymbol{c}_{k-j}$, respectively. The previous output of the block and the previous LSTM state are depicted as $\boldsymbol{h}_{k-j-1}$ and $\boldsymbol{c}_{k-j-1}$, respectively. The LSTM block contains four gate units: the forget gate, input gate, input candidate gate, and output gate. Their outputs are denoted as: $\boldsymbol{f}_{k-j}$, $\boldsymbol{i}_{k-j}$, $\boldsymbol{g}_{k-j}$ and $\boldsymbol{o}_{k-j}$, respectively, and are calculated as follows:

$$\boldsymbol{f}_{k-j} = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_{k-j} + \boldsymbol{R}_f \boldsymbol{h}_{k-j-1} + \boldsymbol{b}_f\right) \tag{9}$$

$$\boldsymbol{i}_{k-j} = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_{k-j} + \boldsymbol{R}_i \boldsymbol{h}_{k-j-1} + \boldsymbol{b}_i\right) \tag{10}$$

$$\boldsymbol{g}_{k-j} = \tanh\left(\boldsymbol{W}_g \boldsymbol{x}_{k-j} + \boldsymbol{R}_g \boldsymbol{h}_{k-j-1} + \boldsymbol{b}_g\right) \tag{11}$$

$$\boldsymbol{o}_{k-j} = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_{k-j} + \boldsymbol{R}_o \boldsymbol{h}_{k-j-1} + \boldsymbol{b}_o\right) \tag{12}$$

where $\boldsymbol{W}_f$, $\boldsymbol{R}_f$, $\boldsymbol{W}_g$, $\boldsymbol{R}_g$, $\boldsymbol{W}_i$, $\boldsymbol{R}_i$,$\boldsymbol{W}_o$, $\boldsymbol{R}_o$ are the recurrent weight and input weight matrices, $\boldsymbol{b}_f$, $\boldsymbol{b}_g$. $\boldsymbol{b}_i$, $\boldsymbol{b}_o$, are the bias vectors.

The output of the memory block and the current state of the LSTM block at time $k-j$ are calculated as follows:

$$\boldsymbol{h}_{k-j} = \boldsymbol{o}_{k-j} * \tanh\left(\boldsymbol{c}_{k-j}\right) \tag{13}$$

$$\boldsymbol{c}_{k-j} = \boldsymbol{f}_{k-j} * \boldsymbol{c}_{k-j-1} + \boldsymbol{g}_{k-j} * \boldsymbol{i}_{k-j} \tag{14}$$

where $*$ denotes element wise product, σ is the logistic sigmoid function.

In Equations (9)–(14), the dimensions of the input weight matrices are $N_u \times n_i$, the recurrent weight matrices are $N_u \times N_u$ and the bias vector is $N_u \times 1$; where n_i- is the number of the input time-dependent variable (in this study its 1), N_u is the number of LSTM cells.

3. Case Study

The hydropower plant "Iron Gate 2" is the downstream step of the system working in coupled mode with the hydropower plant "Iron Gate 1". It was built in 1984 on the Danube River, 80 km downstream of Iron Gate 1. In the main course of the Danube, the main hydropower plant is a joint structure composed of two identical connected parts, namely, the state border of Romania and Serbia, an overflow dam, a non-overflow embankment dam, single step ship lock and an additional (Serbian) power plant. The overflow dam is in the middle of the profile with seven spillway bays, each 28 m wide. The cross-sections of the spillways and weirs have trapezoidal shapes. Tangential displacements of the overflow dam were recorded using telemetered inverted pendulums installed on spillway bays 1, 4, and 7.

The variations in downstream and upstream water levels, the mean daily air temperature, and the measured values of tangential displacement at the first, fourth, and seventh spillways from March 2011 to June 2023 are shown in Figures 3–8, respectively.

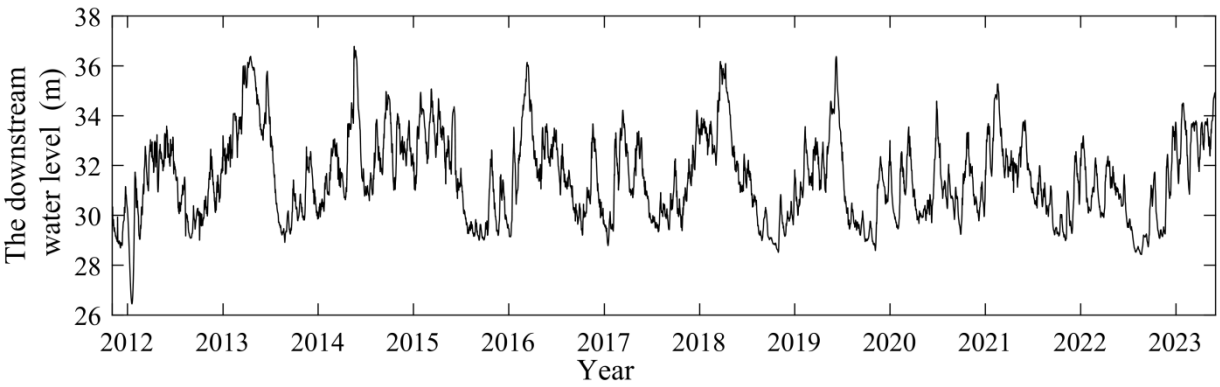

Figure 3. The downstream water level.

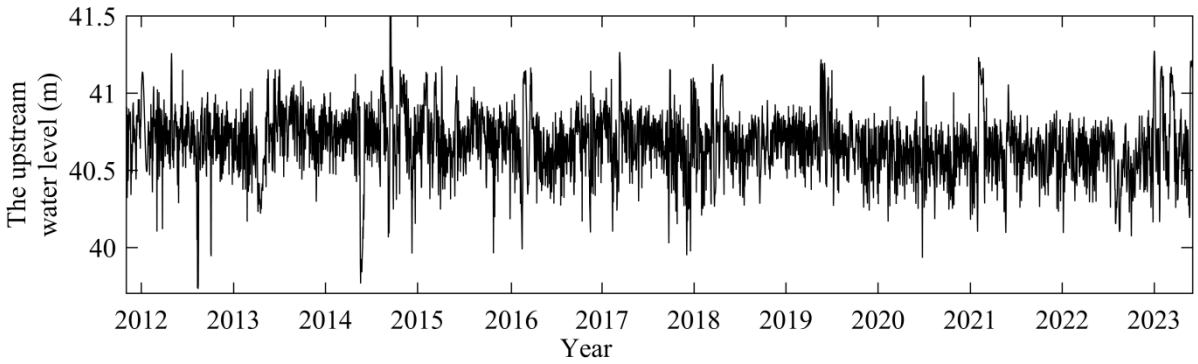

Figure 4. The upstream water level.

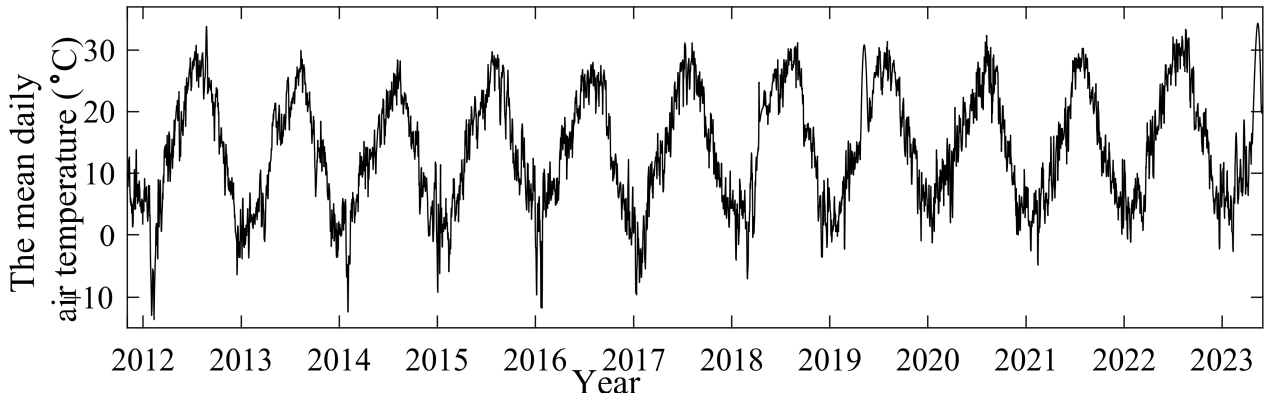

Figure 5. The mean daily air temperature.

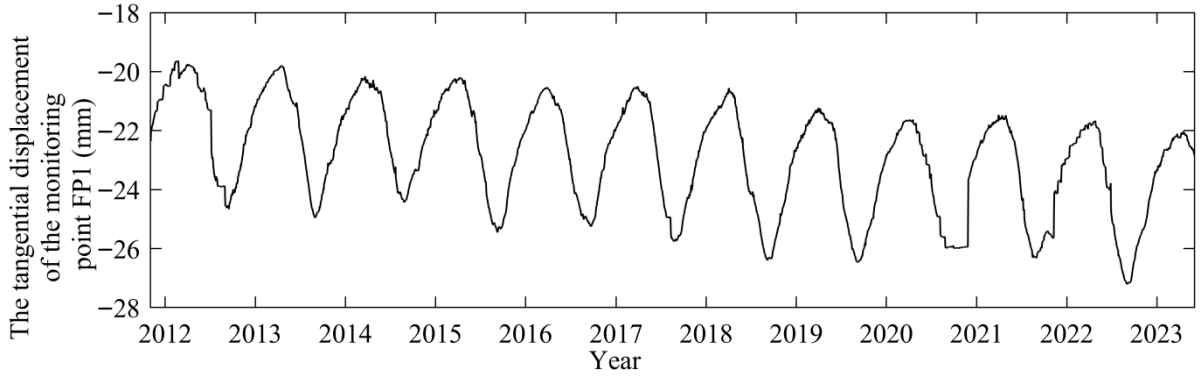

Figure 6. The measured values of tangential displacement at the first spillway.

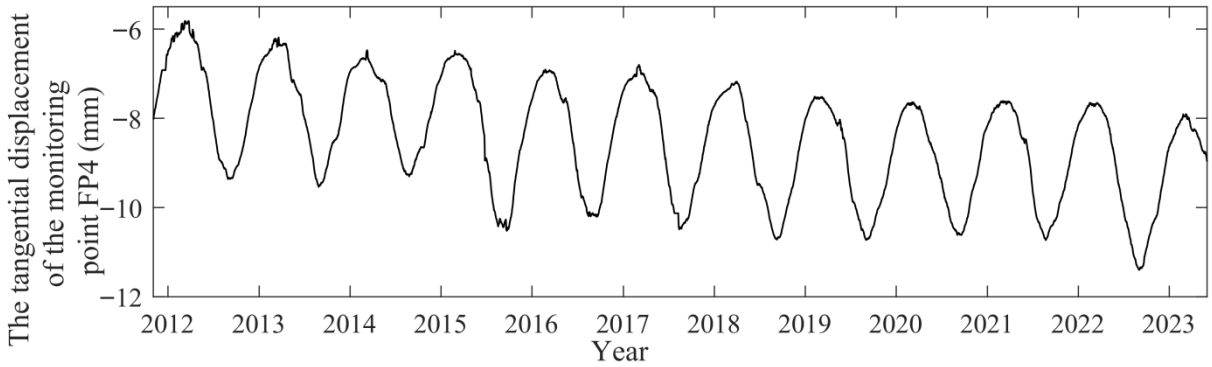

Figure 7. The measured values of tangential displacement at the fourth spillway.

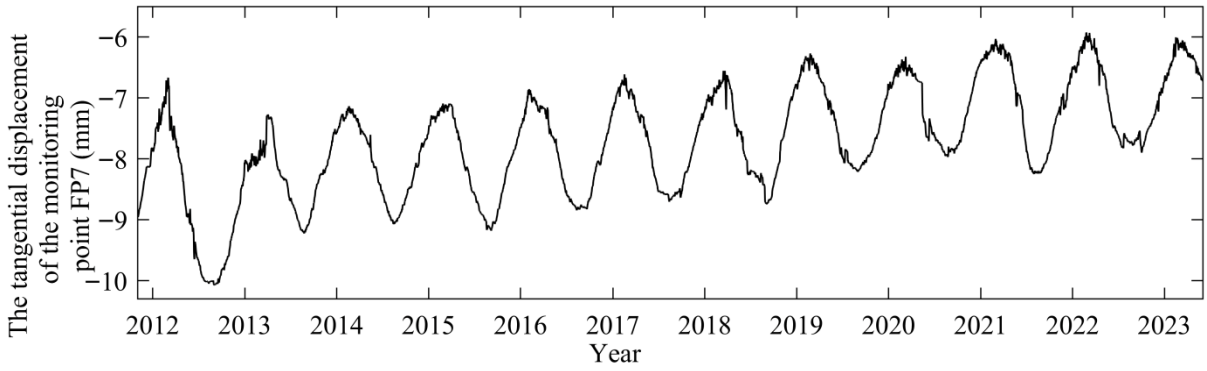

Figure 8. The measured values of tangential displacement at the seventh spillway.

4. Results

Deep NAR-LSTM prediction models with transfer learning for tangential dam displacement were developed for monitoring points FP1, FP4, and FP7 of the spillway dam. The model for predicting dam displacement includes an input sequence layer, two LSTM layers, dropout layers, a fully connected layer, and a regression output layer. The sequence of the input variable was of length (n_{io}) and was formulated as $(y_i(k-n_{io}),\ldots,y_i(k-j),\ldots,y_i(k-1))$. The aim of the proposed NAR-LSTM-based model was to predict the next values of the target time series $y_{ip}(k)$.

The tangential displacements collected from three monitoring points, FP1, FP4, and FP7, of the spillway dam were used for modeling. The dataset consisted of 4230 displacement measurements collected from March 2011 to June 2023. Each time series from the monitoring points was divided into three subsets: 2961 data samples (the first 70% of the dataset) were used for training, 634 samples were assigned to the validation set, and 635 samples (the last 15% of the dataset) were used for testing.

Spline interpolation was applied to estimate the missing data. The time series data were normalized before model development using min–max normalization as follows:

$$y_n = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \tag{15}$$

where y_n represents the normalized value, y denotes the original value, $y_{\min}$ and $y_{\max}$ denotes the maximum and minimum values of the time series samples, respectively.

The training and validation sets of dam displacements for all monitoring points were decomposed using STL decomposition. Numerous simulations were conducted to determine appropriate parameters of the STL algorithm for the decomposition of the dam displacement time series. The periodogram was used to estimate the dominant periods of the dam displacement time series. The seasonal components of all displacement time series showed an approximately one-year cycle (352 days). The dominant periods of the air temperature time series and the upstream water level time series are 352 days. The choice of parameter values, acceptable for all tangential displacement time series, was performed based on the simulation results.

The smoothing parameters for the trend component were chosen to satisfy the following condition [43]:

$$n_{(t)} \geq \frac{1.5 n_{(p)}}{1 - \frac{1}{n_{(s)}}} \tag{16}$$

The smoothing parameters for the low-pass filter were set to the smallest odd number greater than or equal to the number of samples in each cycle of the seasonal component ($n_{(s)}$).

The parameter values of the STL decomposition for points FP1, FP4, and FP7 are presented in Table 1.

Table 1. The parameter values of the STL algorithm.

Parameter	Value
$n_{(p)}$	352
$n_{(s)}$	125
$n_{(t)}$	535
$n_{(l)}$	353

The STL decomposition of the dam displacement was performed using Python (v. 3.10, Python Software Foundation, Wilmington, DE, USA) and the statsmodels module (v. 0.14.0,

statsmodels developers). The decomposition results of the tangential displacement time series for three monitoring points, FP1, FP4, and FP7, are shown in Figures 9–11, respectively.

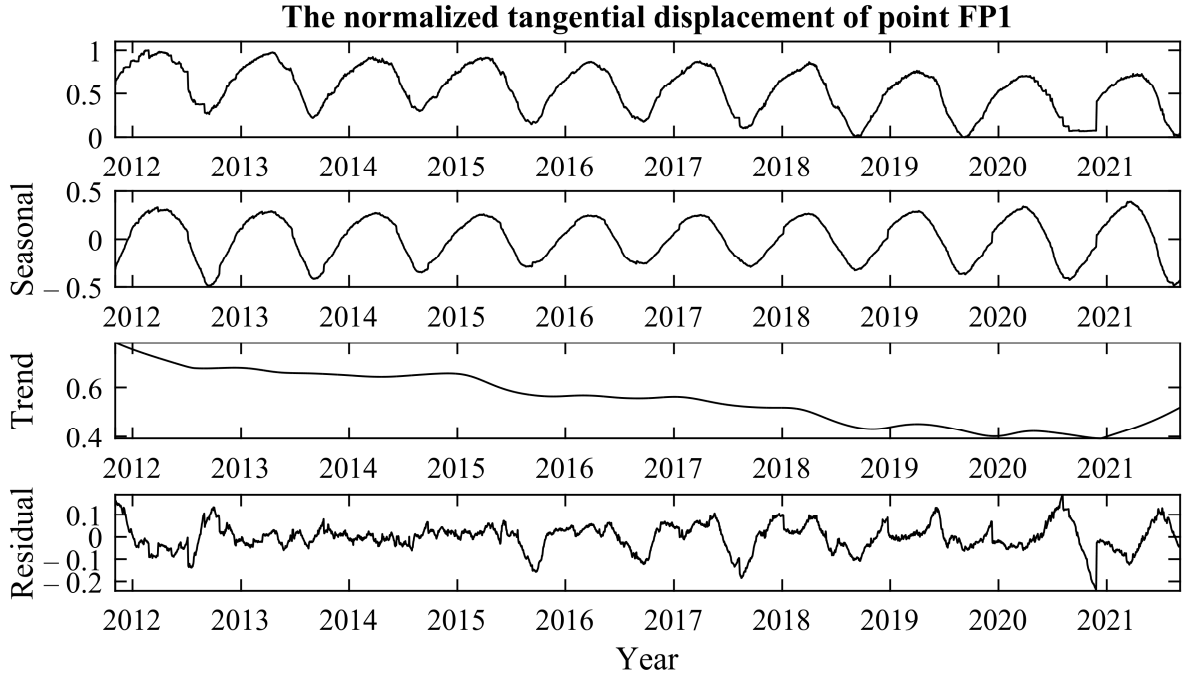

Figure 9. The decomposition results of the tangential displacement time series for monitoring point FP1.

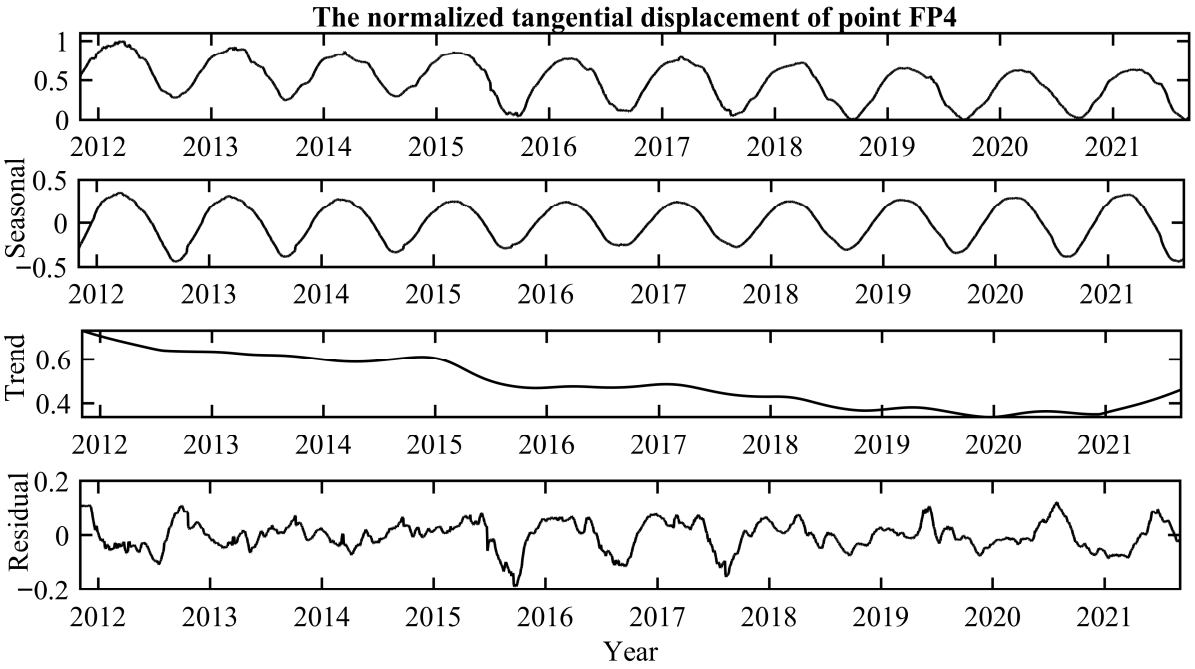

Figure 10. The decomposition results of the tangential displacement time series for monitoring point FP4.

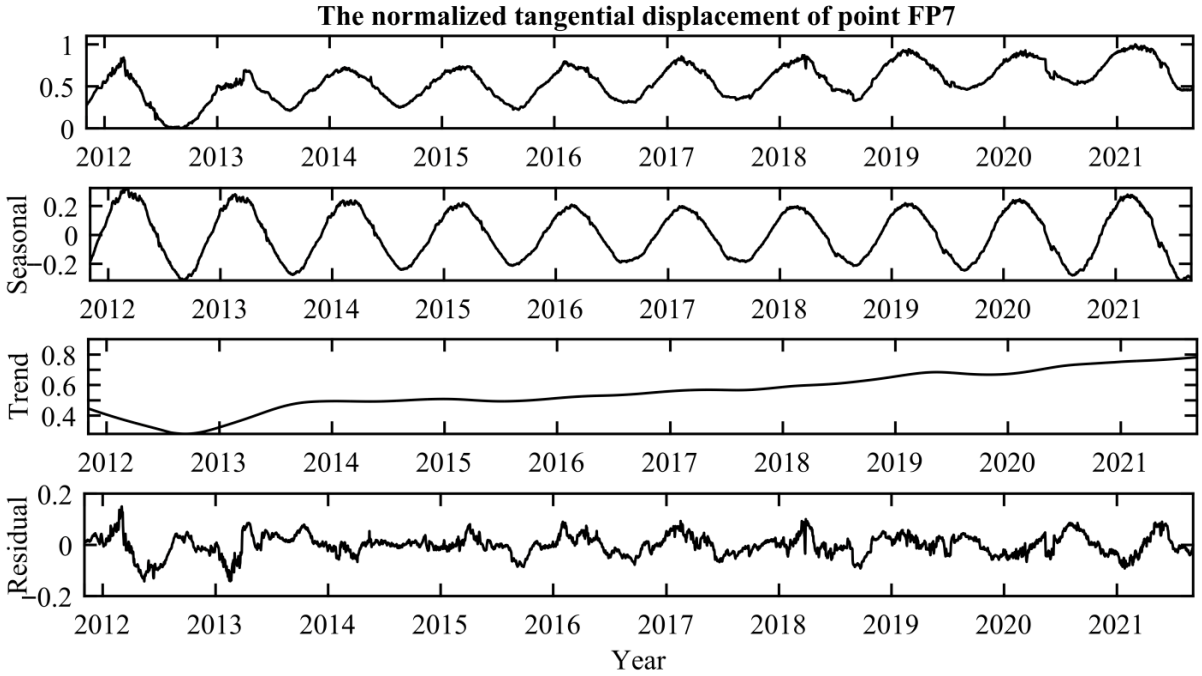

Figure 11. The decomposition results of the tangential displacement time series for monitoring point FP7.

The DTW distances between the target dataset and potential source time series for the dam monitoring points are presented in Table 2. For each measuring point, the source dataset with the smallest DTW distance was selected to train the base model. Based on these results, the time series composed of the seasonal ($S_4(k)$), trend ($T_4(k)$) components of measuring point FP4 and the residual ($R_7(k)$) component of FP7 exhibited the smallest DTW distance of 57.143 from measuring point FP1; it was, therefore, selected as the source dataset for FP1. Similarly, the tangential displacement time series of monitoring point FP1 ($S_{1n} + T_{1n} + R_{1n}$) was selected as the source dataset for FP4, as it yielded the minimal DTW distance of 65.829. Finally, the time series composed of the seasonal ($S_1(k)$), trend ($T_1(k)$) components of FP1 and the residual ($R_4(k)$) component of FP4 showed the smallest DTW distance of 386.891 from measuring point FP7, thus serving as its source dataset.

Table 2. The DTW measure between the target dataset and the potential source time series at the measuring points of the dam.

Potential Source Time Series for FP1 Point	DTW Distance from Time Series y_{1n}	Potential Source Time Series for FP4 Point	DTW Distance from Time Series y_{4n}	Potential Source Time Series for FP7 Point	DTW Distance from Time Series y_{7n}
$S_{7n} + T_{7n} + R_{7n}$	392.045	$S_{1n} + T_{1n} + R_{1n}$	65.829	$S_{4n} + T_{4n} + R_{4n}$	471.924
$S_{4n} + T_{4n} + R_{4n}$	65.829	$S_{4n} + T_{4n} + R_{4n}$	471.924	$S_{1n} + T_{1n} + R_{1n}$	392.046
$S_{4n} + T_{4n} + R_{7n}$	57.143	$S_{1n} + T_{1n} + R_{7n}$	96.53	$S_{1n} + T_{1n} + R_{4n}$	386.891
$S_{4n} + T_{7n} + R_{4n}$	365.994	$S_{1n} + T_{7n} + R_{1n}$	431.461	$S_{1n} + T_{4n} + R_{1n}$	444.046
$S_{7n} + T_{4n} + R_{4n}$	129.399	$S_{7n} + T_{1n} + R_{1n}$	90.18	$S_{4n} + T_{1n} + R_{1n}$	420.197
$S_{7n} + T_{7n} + R_{4n}$	406.254	$S_{7n} + T_{7n} + R_{1n}$	455.204	$S_{4n} + T_{4n} + R_{1n}$	475.817
$S_{7n} + T_{4n} + R_{7n}$	121.104	$S_{7n} + T_{1n} + R_{7n}$	140.015	$S_{4n} + T_{1n} + R_{4n}$	416.178
$S_{4n} + T_{7n} + R_{7n}$	342.842	$S_{1n} + T_{7n} + R_{7n}$	416.08	$S_{1n} + T_{4n} + R_{4n}$	436.669

n in the subscript denotes the normalized value.

Selecting an appropriate combination of hyperparameters for the proposed NAR-LSTM model directly influences its performance. The hyperparameters of the LSTM model for time series forecasting are the length of the input sequence (n_{io}), number of epochs, batch size, optimizer, learning rate, number of LSTM layers and units in each LSTM layer, and dropout rate.

The main techniques for tuning the hyperparameters of LSTM based models include trial and error, grid search, random search, Bayesian optimization, and nature-inspired methods [49]. In this paper, a grid search was applied to optimize the hyperparameters. The optimal length of the input sequence (Equation (7)) was crucial, as it significantly affected the performance of the NAR predictive model. In addition, the optimal number of cells in the LSTM layers substantially influenced the ability to approximate the unknown function (Equation (7)). For all dam displacement prediction models, the number of epochs was set to 100, the batch size to 20, and the learning rate to 0.001. The architecture consisted of two LSTM layers with a dropout rate of 0.2. The Adam optimizer was employed for training. The hyperparameters were set as follows: the length of the input sequence $\in \{2, 3, 4, 5\}$ and the number of cells in the LSTM layers $\in \{8, 16, 24, 32\}$. The optimal parameter values for the model were determined separately for each monitoring point. The optimized values of the model hyperparameters for points FP1, FP4, and FP7 are shown in Table 3.

Table 3. The optimized values of the model hyperparameters for points FP1, FP4, and FP7.

Measuring Point	Length of the Input Sequence	Number of Cells in the First LSTM Layer	Number of Cells in the Second LSTM Layer
FP 1	4	24	16
FP 4	3	24	8
FP 7	5	16	8

The base models for predicting the tangential displacements of monitoring points FP1, FP4, and FP7 were trained on source domain time series data generated and selected based on Table 2. During the transfer learning phase of the dam displacement prediction model, it was possible to apply different approaches to parameter adjustment for the target domain. In the first strategy, shallow layers were fully frozen, allowing only the parameters of the deeper layers to adapt to the target domain data. In the second strategy, lower learning rates were assigned to the shallow layers so that they could adapt to the target domain while preserving the knowledge gained during the pre-training phase. In this study, during the fine-tuning phase of the models at measuring points FP1, FP4, and FP7, a learning rate of 0.0001 was assigned to the first LSTM layer, while the second LSTM layer was assigned a learning rate of 0.0005. This allowed the updating of the recurrent and input weight matrices, as well as the bias vectors of the LSTM layers during transfer learning, to adapt to the target task while preserving the knowledge acquired during the pre-training phase.

To analyze the influence of transfer learning, NAR-LSTM models were developed without fine-tuning for each measurement point and trained solely on target time series data.

The correlation coefficient (R), the mean square error (MSE), and the mean absolute error (MAE) were selected to evaluate the prediction accuracy of the developed NAR-LSTM dam displacement models and were calculated as follows:

$$\mathrm{R} = \frac{\sum_{k=n_{io}+1}^{m} \left(y_{ip}(k) - \overline{y}_{ip}\right)\left(y_i(k) - \overline{y}_i\right)}{\sqrt{\sum_{k=n_{io}+1}^{m} \left(y_{ip}(k) - \overline{y}_{ip}\right)^2 \sum_{k=n_{io}+1}^{m} \left(y_i(k) - \overline{y}_i\right)^2}} \tag{17}$$

$$\mathrm{MSE} = \frac{1}{N} \sum_{k=n_{io}+1}^{m} \left(y_{ip}(k) - y_i(k)\right)^2 \tag{18}$$

$$\mathrm{MAE} = \frac{1}{N} \sum_{k=n_{io}+1}^{m} \left|y_{ip}(k) - y_i(k)\right| \tag{19}$$

where y_{ip} and y_i are the predicted and measured values for the ith dam point; $\overline{y}_{ip}$ and $\overline{y}_i$ denote their average values, respectively; and m denotes the number of observations in the data set.

The performance parameters of the NAR-LSTM models without fine-tuning and the NAR-LSTM models with fine-tuning for forecasting the tangential displacement at measuring points FP1, FP4, and FP7 are presented in Table 4. Figures 12–14 show the measured and modeled values of the tangential displacement at points FP1, FP4, and FP7 in the test sets, generated by both the NAR-LSTM model without fine-tuning and the NAR-LSTM model with fine-tuning.

Table 4. Performance parameters of the NAR-LSTM models for predicting the tangential displacement of the dam measuring points.

MODEL	Measuring Point	Data Set	R	MSE/mm^2	MAE/mm
NAR-LSTM with fine-tuning	FP 1	Training	0.9958	0.0057	0.0629
		Test	0.9926	0.0103	0.0871
	FP4	Training	0.9988	0.0019	0.0393
		Test	0.9915	0.0054	0.0608
	FP7	Training	0.9997	0.0010	0.0265
		Test	0.9912	0.0075	0.0450
NAR-LSTM without fine-tuning	FP 1	Training	0.9895	0.0458	0.1772
		Test	0.9811	0.0792	0.1887
	FP4	Training	0.9814	0.0156	0.0944
		Test	0.9751	0.0297	0.1488
	FP7	Training	0.991	0.0030	0.0380
		Test	0.9899	0.0105	0.0816

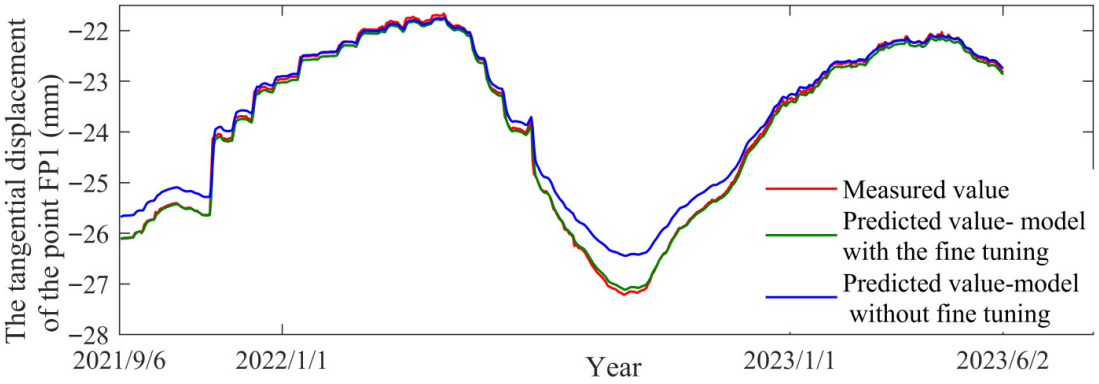

Figure 12. The measured and modeled values of the tangential displacement at measuring point FP1 in the test set.

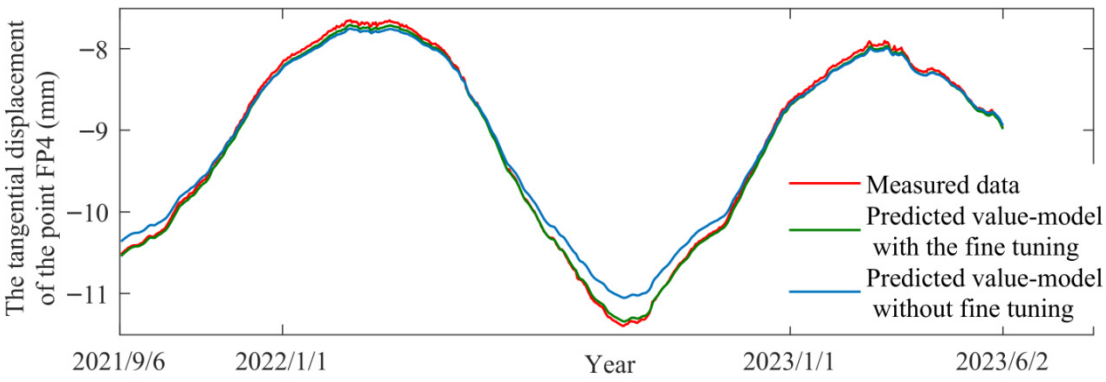

Figure 13. The measured and modeled values of the tangential displacement at measuring point FP4 in the test set.

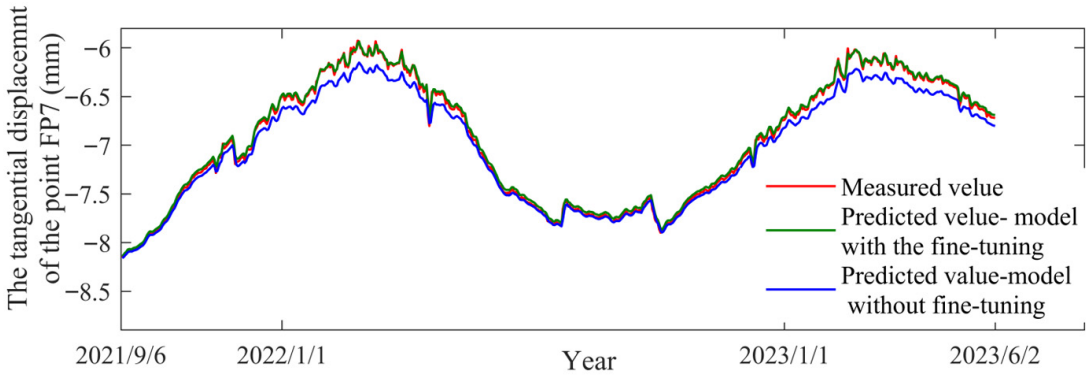

Figure 14. The measured and modeled values of the tangential displacement at measuring point FP7 in the test set.

As shown in Table 4 and Figures 12–14, the fine-tuned NAR-LSTM models significantly outperformed the models without fine-tuning. The improvements in MSE and MAE were expressed as percentage reductions relative to the values obtained from the models without fine-tuning. For the training sets, the MSE decreased to 0.0057, 0.0019, and 0.0010 for FP1, FP4, and FP7, corresponding to reductions of 87.55%, 87.82%, and 66.67%, while the MAE decreased to 0.0629, 0.0393, and 0.0265, corresponding to reductions of 64.51%, 58.38%, and 30.26%. For the test sets, the MSE decreased to 0.0103, 0.0054, and 0.0075, corresponding to reductions of 87.0%, 81.82%, and 28.57%, while the MAE decreased to 0.0871, 0.0608, and 0.0450, corresponding to reductions of 53.84%, 59.18%, and 44.85%. Across the three monitoring points, the average reductions were 80.68% and 65.79% for MSE on the training and test sets, and 51.05% and 52.62% for MAE, respectively. The prediction accuracy of the target nonlinear functions was improved by transferring knowledge from the pre-trained model.

The performance of the proposed model was compared with that of several Machine Learning (ML) models, including Random Forest, Support Vector Machine, and Multi-Layer Perceptron. A grid search was conducted to optimize the hyperparameters of the ML models at each measurement point.

The optimization process for the RF model involved tuning the total number of trees in the ensemble (*n_estimators*) and the minimum number of observations required to split a node (*min_ss*). The hyperparameters were set as *n_estimators* ∈ {50, 100, 150, 200} and *min_ss* ∈ {2, 3, 4, 5, 6}. For the SVR model (which utilized the RBF kernel), the kernel width (σ), regularization parameter (C), and tolerance parameter (ε) were optimized. The hyperparameters were set as $\sigma \in$ {3, 7, 10, 15, 20}, C ∈ {4, 10, 100, 600}, and $\varepsilon \in$ {0.002, 0.005, 0.05, 0.1}. The MLP model involved fine-tuning the number of hidden layers (*Nl*) and the number of hidden neurons (*Nh*), which varied from 1 to 3 and from 5 to 15, respectively. The neurons in the hidden layers used bipolar sigmoid activation functions, whereas those in the output layer employed a Rectified Linear Unit (ReLU) activation function. The optimal values of the hyperparameters and performance parameters of the ML models for predicting tangential displacement are presented in Table 5.

As shown in Table 5, the RF model demonstrated the best performance among the implemented machine learning models for predicting dam displacement. However, as shown in Table 4, the proposed model outperformed all the implemented ML models, including the RF.

Table 5. The optimal values of the hyperparameters and performance parameters of the ML models for predicting tangential displacement.

MODEL	Measuring Point	Data Set	R	MSE/mm^2	MAE/mm
SVR	FP1 $C = 10\ \varepsilon = 0.005\ \sigma = 15$	Training	0.947	0.213	0.284
		Test	0.936	0.2776	0.311
	FP4 $C = 100\ \varepsilon = 0.002\ \sigma = 7$	Training	0.9534	0.1998	0.2347
		Test	0.9504	0.2458	0.2671
	FP7 $C = 100\ \varepsilon = 0.005\ \sigma = 10$	Training	0.9365	0.2541	0.2768
		Test	0.9269	0.2981	0.2976
RF	FP 1 *n_estimators* = 150 *min_ss* = 4	Training	0.9800	0.0166	0.0994
		Test	0.9751	0.0781	0.1439
	FP4 *n_estimators* = 100 *min_ss* = 3	Training	0.9814	0.0156	0.0944
		Test	0.9721	0.0337	0.1518
	FP7 *n_estimators* = 200 *min_ss* = 4	Training	0.989	0.0299	0.0380
		Test	0.9724	0.0432	0.1534
MLP	FP 1 *Nl* = 1 *Nh* = 13	Training	0.9501	0.1991	0.2562
		Test	0.9451	0.2001	0.3484
	FP 4 *Nl* = 1 *Nh* = 16	Training	0.9226	0.3156	0.5747
		Test	0.9176	0.7345	0.9473
	FP 7 *Nl* = 1 *Nh* = 11	Training	0.9375	0.4859	0.6758
		Test	0.9247	0.6748	0.9956

5. Discussion

The performance of the model presented in this study can be compared with that of deep learning models proposed in the literature for concrete gravity dam displacement prediction. Overall, compared with the models reported in the literature, the proposed NAR-LSTM model with fine-tuning achieved comparable predictive performance for concrete gravity dam displacement. Several models that have been proposed in the literature over the past few years for dam behavior prediction were developed by combining CNN layers to extract features from environmental data and recurrent layers to capture the time-dependent characteristics of the time series. Zhang et al. [33] used DenseNet and LSTM networks to predict the deformation of concrete gravity dams. The inputs of the proposed model were water level features, temperature features, and aging features, analyzed and selected using a random forest. The performance of the model for predicting the displacement of four measurement points on a concrete gravity dam was presented. The correlation coefficient, MAE, and RMSE of the datasets were in the ranges of 0.991–0.995, 0.022–0.157, and 0.030–0.195, respectively.

Ren et al. [25] proposed an encoder–decoder dam displacement prediction model that combined an LSTM network with an attention mechanism. The displacements of four monitoring points on the concrete gravity dam were used to assess the validity of the proposed model. The values of the correlation coefficient, MAE, and RMSE for the datasets were in the ranges of 0.899–0.928, 0.313–1.292, and 0.39–1.588, respectively.

Deep transfer learning models have been used mainly for predicting the displacement of concrete arch dams. Chen et al. [35] developed a combined model that included one CNN layer and two LSTM layers to predict the deformation of an arch dam with insufficient observational data. The monitoring data of another dam were used as source data for

transfer learning. The inputs to the source and target models were feature components obtained from dam displacement decomposition. The coefficient of determination, MAE, RMSE, and MAPE were shown for the four dam points in the figure. The coefficient of determination of the models was above 0.995.

6. Conclusions

The assessment of dam safety and its affiliated structures is of paramount importance. Dam displacement prediction is a fundamental component and a key indicator of dam structural health assessment. The dam monitoring data are mostly nonlinear, nonstationary time series. The complexity of dam displacement time series, which primarily results from their nonstationary nature, often limits the performance of forecasting models based on classical statistical methods and machine learning techniques.

This paper proposes the use of transfer learning and fine-tuning strategies with deep LSTM recurrent neural networks for predicting concrete gravity dam displacement. The NAR-LSTM model was first pre-trained using the source domain, and then the target domain was applied to fine-tune the LSTM model to obtain the target dam displacement prediction model.

A new method for generating the source domain was introduced, which is based on STL decomposition and the DTW distance measure. In future research, it would be particularly useful to validate the proposed approach using displacement data collected from monitoring systems of various concrete gravity dams. These data would be particularly useful for training the base model and generating a set of potential source time series.

The proposed NAR-LSTM model with fine-tuning achieved excellent displacement prediction accuracy because fine-tuning all layers significantly enhanced the adaptability of the model to the target domain.

Data-driven models for predicting dam behavior need to be updated to effectively capture dynamic changes influenced by external factors and structural evolution. The proposed approach enables a simple update of the developed models. Target models for measurement points FP1, FP4, and FP7 can serve as pre-trained models and be fine-tuned with new data obtained from the monitoring system to increase the accuracy of displacement prediction.

In this study, we applied the proposed approach to predict dam displacement and found that it provided accurate predictions. We believe that this methodology can also be applied to other engineering time series forecasting problems, both to improve predictive performance when sufficient data are available and to solve the problem of limited target data. While large-scale AI models offer broad contextual capabilities, this study demonstrates that for domain-specific tasks involving spatially and temporally correlated sensor data, specialized architectures such as NAR-LSTM provide superior numerical precision and practical effectiveness.

Author Contributions: Conceptualization, V.Ć. and V.R.; methodology, V.R. and N.M.; software, V.Ć. and B.M.-D.; validation, V.Ć., V.R. and N.M.; formal analysis, V.Ć.; investigation, V.Ć. and V.M.; resources, N.M.; data curation, V.Ć., V.M. and B.M.-D.; writing—original draft preparation, V.Ć. and V.R.; writing—review and editing, V.R. and V.Ć.; visualization, V.Ć., V.M. and B.M.-D.; supervision, V.R. and N.M.; project administration, N.M. and V.R.; funding acquisition, Not applicable. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability Statement: All original contributions presented in this study are included within the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This research was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, contract number 451-03-34/2026-03/200107 (Faculty of Engineering, University of Kragujevac).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, M.; Shen, Y.; Ren, Q.; Li, H. A new distributed time series evolution prediction model for dam deformation based on constituent elements. *Adv. Eng. Inf.* **2019**, *39*, 41–52. [CrossRef]
2. Wei, B.; Yuan, D.; Li, H.; Xu, Z. Combination forecast model for concrete dam displacement considering residual correction. *Struct. Health Monit.* **2019**, *18*, 232–244. [CrossRef]
3. Ren, Q.; Li, M.; Kong, R.; Shen, Y.; Du, S. A hybrid approach for interval prediction of concrete dam displacements under uncertain conditions. *Eng. Comput.* **2023**, *39*, 1285–1303. [CrossRef]
4. Lin, C.; Li, T.; Chen, S.; Lin, C.; Liu, X.; Gao, L.; Sheng, T. Structural identification in long-term deformation characteristic of dam foundation using meta-heuristic optimization techniques. *Adv. Eng. Softw.* **2020**, *148*, 102870. [CrossRef]
5. Žvanut, P. 3D Finite Element Analysis of a Concrete Dam Behavior under Changing Hydrostatic Load: A Case Study. *Materials* **2022**, *15*, 921. [CrossRef]
6. Fenza, G.; Gallo, M.; Loia, V.; Orciuoli, F.; Herrera-Viedma, E. Data set quality in machine learning: Consistency measure based on group decision making. *Appl. Soft. Comput.* **2021**, *106*, 107366. [CrossRef]
7. Bao, Y.; Tang, Z.; Li, H.; Zhang, Y. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Struct. Health Monit.* **2019**, *18*, 401–421. [CrossRef]
8. Zhang, Y.; Tang, Z.; Yang, R. Data anomaly detection for structural health monitoring by multi-view representation based on local binary patterns. *Measurement* **2022**, *202*, 111804. [CrossRef]
9. Salazar, F.; Toledo, M.Á.; Oñate, E.; Suárez, B. Interpretation of dam deformation and leakage with boosted regression trees. *Eng. Struct.* **2016**, *119*, 230–251. [CrossRef]
10. Kang, F.; Liu, J.; Li, J.; Li, S. Concrete dam deformation prediction model for health monitoring based on extreme learning machine. *Struct. Control Health Monit.* **2017**, *24*, e1997. [CrossRef]
11. Chen, B.; Huang, Z.; Bao, T.; Zhu, Z. Deformation early-warning index for heightened gravity dam during impoundment period. *Water. Sci. Eng.* **2021**, *14*, 54–64. [CrossRef]
12. Su, Y.; Weng, K.; Lin, C.; Zheng, Z. An Improved Random Forest Model for the Prediction of Dam Displacement. *IEEE Access* **2021**, *9*, 9142–9153. [CrossRef]
13. Sapankevych, N.I.; Sankar, R. Time Series Prediction Using Support Vector Machines: A Survey. *IEEE Comput. Intell. Mag.* **2009**, *4*, 24–38. [CrossRef]
14. Ranković, V.; Grujović, N.; Divac, D.; Milivojević, N. Development of support vector regression identification model for prediction of dam structural behavior. *Struct. Saf.* **2014**, *48*, 33–39. [CrossRef]
15. Cheng, M.-Y.; Cao, M.-T.; Huang, I.-F. Hybrid artificial intelligence-based inference models for accurately predicting dam body displacements: A case study of the Fei Tsui dam. *Struct. Health Monit.* **2022**, *21*, 1738–1756. [CrossRef]
16. Wang, S.; Xu, C.; Liu, Y.; Wu, B. A spatial association-coupled double objective support vector machine prediction model for diagnosing the deformation behaviour of high arch dams. *Struct. Health Monit.* **2022**, *21*, 945–964. [CrossRef]
17. Su, H.; Wen, Z.; Chen, Z.; Tian, S. Dam safety prediction model considering chaotic characteristics in prototype monitoring data series. *Struct. Health Monit.* **2016**, *15*, 639–649. [CrossRef]
18. Su, H.; Li, X.; Yang, B.; Wen, Z. Wavelet support vector machine-based prediction model of dam deformation. *Mech. Syst. Signal Process.* **2018**, *110*, 412–427. [CrossRef]
19. Mata, J. Interpretation of concrete dam behaviour with artificial neural network and multiple linear regression models. *Eng.Struct.* **2011**, *33*, 903–910. [CrossRef]
20. Stojanovic, B.; Milivojevic, M.; Milivojevic, N.; Antonijevic, D. A self-tuning system for dam behavior modeling based on evolving artificial neural networks. *Adv. Eng. Softw.* **2016**, *97*, 85–95. [CrossRef]
21. Guzmán Sejas, A.M.; Pereira, S.; Mata, J.; Cunha, Á. Developing Statistical and Multilayer Perceptron Neural Network Models for a Concrete Dam Dynamic Behaviour Interpretation. *Infrastructures* **2025**, *10*, 301. [CrossRef]
22. Kang, F.; Li, J.; Zhao, S.; Wang, Y. Structural health monitoring of concrete dams using long-term air temperature for thermal effect simulation. *Eng. Struct.* **2019**, *180*, 642–653. [CrossRef]
23. Ranković, V.; Grujović, N.; Divac, D.; Milivojević, N.; Novaković, A. Modelling of dam behaviour based on neuro-fuzzy identification. *Eng. Struct.* **2012**, *35*, 107–113. [CrossRef]
24. Salazar, F.; Toledo, M.A.; Oñate, E.; Morán, R. An empirical comparison of machine learning techniques for dam behaviour modeling. *Struct. Saf.* **2015**, *56*, 9–17. [CrossRef]

25. Madiniyeti, J.; Chao, Y.; Li, T.; Qi, H.; Wang, F. Concrete Dam Deformation Prediction Model Research Based on SSA–LSTM. *Appl. Sci.* **2023**, *13*, 7375. [CrossRef]
26. Liu, W.; Pan, J.; Ren, Y.; Wu, Z.; Wang, J. Coupling prediction model for long-term displacements of arch dams based on long short-term memory network. *Struct. Control Health Monit.* **2020**, *27*, e2548. [CrossRef]
27. Ren, Q.; Li, M.; Li, H.; Shen, Y. A novel deep learning prediction model for concrete dam displacements using interpretable mixed attention mechanism. *Adv. Eng. Inf.* **2021**, *50*, 101407. [CrossRef]
28. Wen, Z.; Zhou, R.; Su, H. MR and stacked GRUs neural network combined model and its application for deformation prediction of concrete dam. *Expert Syst. Appl.* **2022**, *201*, 117272. [CrossRef]
29. Xu, R.; Liu, X.; Wei, J.; Ai, X.; Li, Z.; He, H. Predicting the Deformation of a Concrete Dam Using an Integration of Long Short-Term Memory (LSTM) Networks and Kolmogorov–Arnold Networks (KANs) with a Dual-Stage Attention Mechanism. *Water* **2024**, *16*, 3043. [CrossRef]
30. Ma, J.; Huang, X.; Wu, H.; Yan, K.; Liu, Y. Bidirectional Gated Recurrent Unit (BiGRU)-Based Model for Concrete Gravity Dam Displacement Prediction. *Sustainability* **2025**, *17*, 7401. [CrossRef]
31. Prakash, G.; Dugalam, R.; Barbosh, M.; Sadhu, A. Recent advancement of concrete dam health monitoring technology: A systematic literature review. *Structures* **2022**, *44*, 766–784. [CrossRef]
32. Ren, Q.; Li, M.; Bai, S.; Shen, Y. A multiple-point monitoring model for concrete dam displacements based on correlated multiple-output support vector regression. *Struct. Health Monit.* **2022**, *21*, 2768–2785. [CrossRef]
33. Hu, J.; Jiang, H.; Li, X. An optimized zonal deformation prediction model for super-high arch dams. *Structures* **2023**, *50*, 758–774. [CrossRef]
34. He, P.; Pan, J.; Li, Y. Long-term dam behavior prediction with deep learning on graphs. *J. Comput. Des. Eng.* **2022**, *9*, 1230–1245. [CrossRef]
35. Pan, J.; Liu, W.; Liu, C.; Wang, J. Convolutional neural network-based spatiotemporal prediction for deformation behavior of arch dams. *Expert Syst. Appl.* **2023**, *232*, 120835. [CrossRef]
36. Li, M.; Li, M.; Ren, Q.; Li, H.; Song, L. DRLSTM: A dual-stage deep learning approach driven by raw monitoring data for dam displacement prediction. *Adv. Eng. Inf.* **2022**, *51*, 101510. [CrossRef]
37. Zhang, Y.; Zhong, W.; Li, Y.; Lifeng, W. A deep learning prediction model of DenseNet-LSTM for concrete gravity dam deformation based on feature selection. *Eng. Struct.* **2023**, *295*, 116827. [CrossRef]
38. Li, Y.; Bao, T.; Gao, Z.; Shu, X.; Zhang, K.; Xie, L.; Zhang, Z. A new dam structural response estimation paradigm powered by deep learning and transfer learning techniques. *Struct. Health Monit.* **2022**, *21*, 770–787. [CrossRef]
39. Chen, X.; Chen, Z.; Hu, S.; Gu, C.; Guo, J.; Qin, X. A feature decomposition-based deep transfer learning framework for concrete dam deformation prediction with observational insufficiency. *Adv. Eng. Inf.* **2023**, *58*, 102175. [CrossRef]
40. Ye, R.; Dai, Q. A relationship-aligned transfer learning algorithm for time series forecasting. *Inf. Sci.* **2022**, *593*, 17–34. [CrossRef]
41. Li, Y.; Bao, T.; Chen, H.; Zhang, K.; Shu, X.; Chen, Z.; Hu, Y. A large-scale sensor missing data imputation framework for dams using deep learning and transfer learning strategy. *Measurement* **2021**, *178*, 109377. [CrossRef]
42. Li, Y.; Zhu, Z.; Kong, D.; Han, H.; Zhao, Y. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowl.-Based Syst.* **2019**, *181*, 104785. [CrossRef]
43. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E.; Terpenning, I. STL: A seasonal-trend decomposition procedure based on loess. *J. Off. Stat.* **1990**, *6*, 3–73.
44. Berndt, D.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the AAAI Workshop on Knowledge Discovery in Databases (KDD-94), Seattle, WA, USA, 31 July 1994.
45. Connor, J.T.; Martin, R.D.; Atlas, L.E. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254. [CrossRef] [PubMed]
46. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 9. [CrossRef]
47. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
48. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef] [PubMed]
49. Bataineh, A.A.; Kaur, D. Immunocomputing-based approach for optimizing the topologies of LSTM networks. *IEEE Access* **2021**, *9*, 78993–79004. [CrossRef]