

Article

# Deep Learning CNN-GRU Method for GNSS Deformation Monitoring Prediction

**Yilin Xie** <sup>1</sup>, **Jun Wang** <sup>1</sup>, **Haiyang Li** <sup>1,\*</sup>, **Azhong Dong** <sup>1</sup>, **Yanping Kang** <sup>1</sup>, **Jie Zhu** <sup>1</sup>, **Yawei Wang** <sup>2,3</sup> and **Yin Yang** <sup>1</sup>

<sup>1</sup> Jiangsu Hydraulic Research Institute, Nanjing 210017, China; yilin.xie@jswater.org.cn (Y.X.); jun.wang@jswater.org.cn (J.W.); azhong.dong@jswater.org.cn (A.D.); yanping.kang@jswater.org.cn (Y.K.); jie.zhu@jswater.org.cn (J.Z.); yin.yang@jswater.org.cn (Y.Y.)

<sup>2</sup> Hunan Engineering Research Center of 3D Real Scene Construction and Application Technology, Changsha 410114, China; grcwongyw2016@whu.edu.cn

<sup>3</sup> The First Surveying and Mapping Institute of Hunan Province, Changsha 410114, China

\* Correspondence: haiyang.li@jswater.org.cn

**Abstract:** Hydraulic structures are the key national infrastructures, whose safety and stability are crucial for socio-economic development. Global Navigation Satellite System (GNSS) technology, as a high-precision deformation monitoring method, is of great significance for the safety and stability of hydraulic structures. However, the GNSS time series exhibits characteristics such as high nonlinearity, spatiotemporal correlation, and noise interference, making it difficult to model for prediction. The Neural Networks (CNN) model has strong feature extraction capabilities and translation invariance. However, it remains sensitive to changes in the scale and position of the target and requires large amounts of data. The Gated Recurrent Units (GRU) model could improve the training effectiveness by introducing gate mechanisms, but its ability to model long-term dependencies is limited. This study proposes a combined model, using CNN to extract spatial features and GRU to capture temporal information, to achieve an accurate prediction. The experiment shows that the proposed CNN-GRU model has a better performance, with an improvement of approximately 45%, demonstrating higher accuracy and reliability in predictions for GNSS deformation monitoring. This provides a new feasible solution for the safety monitoring and early warning of hydraulic structures.



**Citation:** Xie, Y.; Wang, J.; Li, H.; Dong, A.; Kang, Y.; Zhu, J.; Wang, Y.; Yang, Y. Deep Learning CNN-GRU Method for GNSS Deformation Monitoring Prediction. *Appl. Sci.* **2024**, *14*, 4004. <https://doi.org/10.3390/app14104004>

Academic Editors: Bogdan Grecu and Dragos Tataru

Received: 8 April 2024

Revised: 29 April 2024

Accepted: 4 May 2024

Published: 8 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The accelerated progress of global climate change and urbanization makes hydraulic infrastructures increasingly prominent. The hydraulic structures are of paramount importance in safeguarding the safety of people's lives and properties and promoting sustainable economic and social development [1–3]. However, there may be various challenges during their long-term operation, such as geological disasters, land subsidence, and structural deformation. Untimely monitoring and ineffective prediction may lead to serious safety accidents and economic losses.

Many approaches are applied for the monitoring and prediction of hydraulic structures. The traditional approaches rely on engineering surveys [4,5] and geological investigations [6,7]. However, these approaches are limited by their restricted monitoring range and complex processing workflows, making them inadequate for meeting the needs of hydraulic structures [8]. As an all-weather and full-automation piece of technology, GNSS introduces an innovative approach to the deformation monitoring of hydraulic structures [9,10]. It provides real-time and high-precision deformations, facilitating effective analysis and predictions. However, the complex environment around hydraulic structures leads to measurement errors, such as atmospheric delays [11–13] and multipath effects [14–17]. These errors are difficult to process, which adds obstacles to the prediction

of GNSS deformation monitoring. The prediction model necessitates the capability to handle the unmodeled factors and observation noise.

A Kalman filter is a Bayesian-based minimum mean-square estimation method, which lies in adding state constraints to the observation equations, thereby constructing reliable function models and stochastic models [18–20]. However, it is difficult to handle state equations and system errors in complex environments. The grey model is suitable for analyzing and modeling short GNSS time series with limited datasets and incomplete information [21–23]. Nevertheless, it is only suitable for short-term and exponential growth predictions. The Autoregressive Integrated Moving Average (ARIMA) model extracts time series for prediction through the autocorrelation and difference of the dataset [24–26], but it requires the dataset to be stable, essentially capturing linear relationships. The Multiple regression analysis (MRA) model quantitatively describes a dependent variable's linear dependence on multiple independent variables, offering simplicity, high precision, and wide applicability [27–29]. However, it suffers from multicollinearity issues, lacks causal inference capability, and makes assumptions about error terms. The genetic algorithms are inspired by natural evolution theory, dealing with problems lacking mathematical expressions [30–32]. However, it requires special definitions and cannot guarantee the quality of solutions.

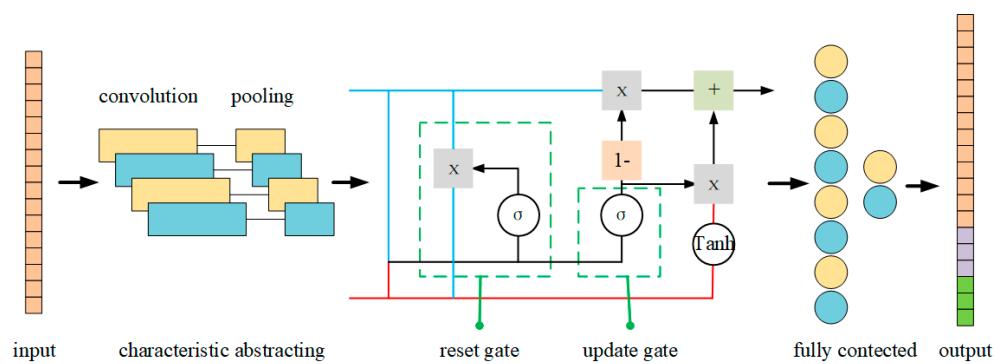
Methods based on deep learning are popular for deformation prediction, which is also applied to the chaotic time series prediction [33,34]. The deep learning methods have powerful feature extraction and recognition capabilities, enabling accurate predictions by learning representations from GNSS time series. The CNN model and Recurrent Neural Networks (RNN) have been applied in the fields of image processing and sequence modeling. Meanwhile, the GRU model represents an improved model of the RNN, adept at capturing long-term dependencies in the dataset [35]. Previous studies have applied CNN for GNSS residual processing [36] and classification [37], GRU for landslide prediction [38] and multipath modeling [39], as well as CNN-GRU for forecasting wind power [40], particulate matter concentrations [41], and the pressure of a concrete dam [42]. The CNN-GRU model integrates the characteristics of CNN and GRU, which has better a performance in predictions of GNSS deformation monitoring.

This study proposes to apply the CNN-GRU model for the GNSS deformation monitoring prediction of hydraulic structures. The CNN model extracts relevant spatial features, further enhancing prediction accuracy, while the GRU model utilizes past datasets to predict future trends. The experiment proves that the CNN-GRU model has a better performance than the CNN and GRU model, with an improvement of about 45%. The proposed method provides a new idea for hydraulic safety management.

The rest of this paper is organized as follows: Section 2 introduces the principles, processes, and details of the methods used in this study. Section 3 validates the effectiveness and superiority of the proposed methods through experiments. Section 4 summarizes the rationality, superiority, and possible issues of the proposed methods. Finally, Section 5 concludes the entire paper, providing analysis and conclusions.

## 2. Method

The proposed method utilizes a comprehensive approach for the prediction of GNSS deformation monitoring in hydraulic structures. It offers a novel framework for feature extraction and long-term sequence modeling, enabling accurate and proactive deformation prediction within hydraulic structures. The workflow integrates the CNN and GRU models to capture spatial-temporal patterns and dependencies, which are presented in Figure 1. The proposed method combines the feature extraction capability of CNN with the temporal modeling ability of GRU, which effectively captures spatial and temporal features in GNSS time series.



**Figure 1.** The flowcharts of the CNN-GRU model.

The steps of the proposed method are as follows:

- (1) Data Preprocessing: firstly, normalize the input data.
- (2) Convolutional Feature Extraction: CNN is employed to extract spatial features from the input data. Typically composed of convolutional layers and pooling layers, CNN performs feature extraction through convolutional operations and reduces the size of feature maps via pooling operations.
- (3) Modeling for GNSS time series: the convolutional features extracted by CNN are inputted into GRU for modeling. Through the update gates and reset gates of GRU, the flow of information is controlled, enabling the network to better capture long-term dependencies within sequences.
- (4) Prediction and Evaluation: the output of GRU is connected to the fully connected layer of CNN to accomplish specific tasks.

The entire CNN-GRU model is trained using labeled data and optimized via a back-propagation algorithm to better fit the training data and generalize new data. The trained model is evaluated using a validation set or test set to assess its performance on unseen data. Various metrics such as accuracy, precision, and recall are utilized to evaluate the model's performance.

### 2.1. Convolutional Feature Extraction

The CNN model is to extract features from input GNSS time series through convolution and pooling processes, and then combine and classify them through multiple layers of neural networks. The CNN model typically consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. This section consists of a convolutional layer and a pooling layer.

The convolutional layer extracts characteristics from input time series through the convolutional process. By sliding the convolutional kernel over the input, it effectively captures local features from different positions, thus comprehensively representing the spatial structure information of the input. Its formula is as follows:

$$a_j^{l+1} = \text{ReLU}\left(\sum_{f=1}^{F^l} K_{j,f}^l a_j^l + b_j^l\right) \quad (1)$$

where  $l$  is the number of layers,  $a_j^{l+1}$  is the  $j$ th characteristic of the  $l + 1$  layer,  $K$  is the convolutional kernel,  $b$  is the bias,  $1 \leq f \leq F^l$ ,  $F^l$  is the characteristic number of  $l$  layer, and  $\text{ReLU}()$  is the activation function. The convolutional layer generates multiple feature maps using multiple convolutional kernels. Each feature map represents distinct features within the input GNSS time series, while the same convolutional kernel uses the same weights at different positions.

The pooling layer downsamples the feature maps of the convolutional layer, thereby reducing their size and subsequently lowering the computational complexity of the model.

The proposed method employs the max-pooling function to downsample the feature maps, as shown in the following formula:

$$x_i^{l+1} = \max_{k=1}^r (x_{i+k}^l) \quad (2)$$

where  $x_i^{l+1}$  is the  $i$ th neuron of the  $l + 1$  layer,  $x_{i+k}^l$  is the  $i + k$  neuron of the  $l$  layer, and  $1 \leq k \leq r$ ,  $r$  is the size of the pooling layer. The max-pooling function reduces the dimensionality of the dataset in the preceding layer by retaining the most significant features. Additionally, the pooling layer alleviates overfitting issues and enhances the generalization ability of the proposed model.

## 2.2. Modeling for GNSS Time Series

The convolutional features extracted by CNN are inputted into GRU for modeling. Similar to the Long Short-Term Memory (LSTM) model, the GRU model is an evolution of the RNN model, aimed at addressing the issues of vanishing gradients and long-term dependencies inherent in the traditional RNN model.

The GRU model includes an update gate and reset gate, while the LSTM model consists of a forget gate, input gate, and output gate. Due to that, the GRU model has fewer parameters with a simpler structure. This facilitates easier convergence during training for the GRU model and enhances its generalization performance on smaller datasets. As shown in Figure 1, the update gate is to update the dataset, while the reset gate is to reset the dataset. The formula of the update gate is as follows:

$$z_t = \text{sigm}(W_{xz}x_t + W_{hz}h_{t-1}) \quad (3)$$

where  $z_t$  is the update gate,  $\text{sigm}()$  is the sigmoid function,  $x_t$  is the input dataset and  $W_{xz}$  denotes its weight matrix,  $h_{t-1}$  is the hidden status of the previous epoch, and  $W_{hz}$  denotes its weight matrix. The update gate determines the information from the previous epoch that needs to be passed on to the future. This mechanism is highly potent as it decides whether to retain all information from the past, thereby mitigating the risk of gradient vanishing.

The reset gate serves a specific function, deciding the criteria for filtering previous state information, which effectively captures short-term dependencies in the dataset after the CNN method. The formula of the reset gate is as follows:

$$r_t = \text{sigm}(W_{xr}x_t + W_{hr}h_{t-1}) \quad (4)$$

where  $r_t$  is the reset gate,  $W_{xr}$  denotes its weight matrix of  $x_t$ , and  $W_{hr}$  denotes its weight matrix of  $h_{t-1}$ . The reset gate influences the model's retention of historical information. If the value of the reset gate is close to 0, the model will ignore the previous hidden state.

The GRU model is a forward-propagating neural network, and a memory unit is designed to store the information from the past epochs. Based on Formulas (3) and (4), the memory unit is shown in Formula (5).

$$c_t = \tanh[W_{xc}x_t + r_t \odot W_{hc}h_{t-1}] \quad (5)$$

where  $c_t$  is the current memory unit,  $W_{xc}$  denotes the weight matrix of  $x_t$  in the hidden state  $h_t$ ,  $W_{hc}$  denotes the weight matrix of  $h_{t-1}$  in the hidden state  $h_t$ , and  $\odot$  denotes the Hadamard product. Based on the above, the final memory unit of the current epoch is shown in Formula (6).

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot c_t \quad (6)$$

where  $h_t$  is the final memory unit, which stores the current information and sends it to the next epoch.

### 2.3. Prediction and Evaluation

The output of GRU is connected to the fully connected layer of CNN to accomplish specific tasks. The fully connected layer integrates features extracted from convolutional and pooling layers and maps them to the GRU model. In this layer, features extracted by convolution and pooling layers are flattened and connected to a fully connected layer, which ultimately outputs the prediction results. The mathematical representation of the fully connected layer is as follows

$$a^l = \sigma(z^l) = \sigma(K^l a^{l-1} + b^l) \quad (7)$$

where  $l$  is the number of layers,  $a^l$  is the output layer,  $a^{l-1}$  is the output of the previous layer,  $K$  is the convolutional kernel,  $b$  is the bias, and  $\sigma(\cdot)$  is the activation function. The fully connected layer integrates features from the GRU model, which has better performance in the prediction of GNSS time series.

The entire CNN-GRU model is trained using a labeled dataset and optimized via a backpropagation algorithm to better fit the training data and generalize to a new dataset. The trained model is evaluated using a validation set or test set to assess its performance on unseen data. The proposed method sets the Mean Squared Error (MSE) as the evaluation, for which the formula is as follows:

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i^{pred} - y_i^{test})^2 \quad (8)$$

where  $m$  is the length of the GNSS time series,  $y_i^{pred}$  denotes the  $i$ th element of the prediction, and  $y_i^{test}$  denotes the corresponding observation.

The CNN-GRU model offers a powerful combination of multi-level feature extraction capabilities from CNN and the ability to capture long-term dependencies in sequential data through GRU. By integrating CNN for spatial feature extraction and GRU for modeling, the proposed method achieves strong adaptability, parameter-sharing benefits, and end-to-end training efficiency, making it highly effective for GNSS time series prediction.

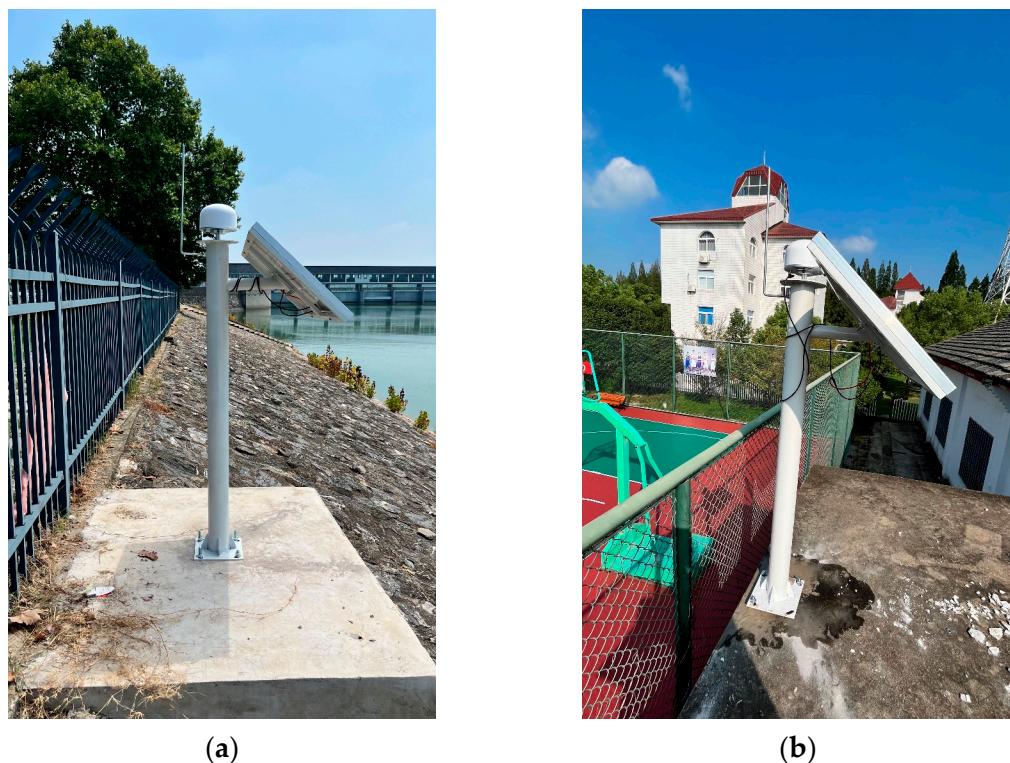
### 3. Experiment

The experiment was conducted at Sanhe Sluice of Hongze Lake, Jiangsu, China, for which the configure parameters are as shown in Table 1. The monitoring site and the reference site are shown in Figure 2.

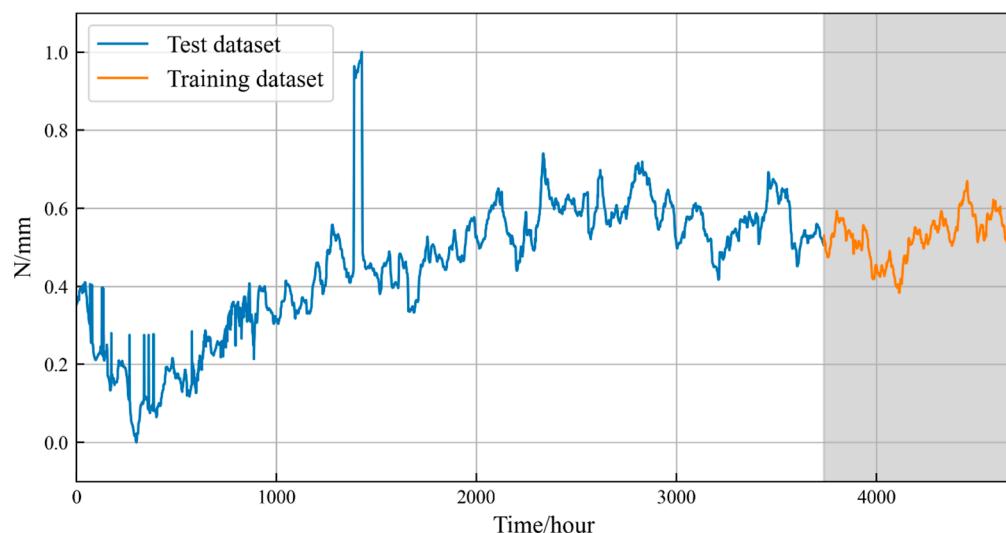
**Table 1.** The configure parameters of the feasibility experiment.

Configure	Parameter
The experiment period	10 September 2023–29 March 2024
GNSS systems	BDS(B1I, B2I), Galileo(E1B/C, E5b), GPS(L1C/A, L2C)
Sampling frequency	1 Hz
Ambiguity resolution method	MLAMBDA
Multipath error model	Stellar day filter in observational domain
Troposphere method	Saastamoinen Model + Random Walk
Ionosphere method	Broadcast model
Intervals of outputs	1 h
Smoothing filter method	Rauch–Tung–Striebel Smoother Filtering Method

This study validates the effectiveness of the proposed method using a North (N)-direction time series as an example. We use 80% of the time series as the training set and the remaining 20% as the validation set. A 24 h extrapolation is provided at the end for prediction. The partition of the GNSS time series is shown in Figure 3.



**Figure 2.** (a) The monitoring station installed on the dam; (b) The base station installed on the roof of a building.

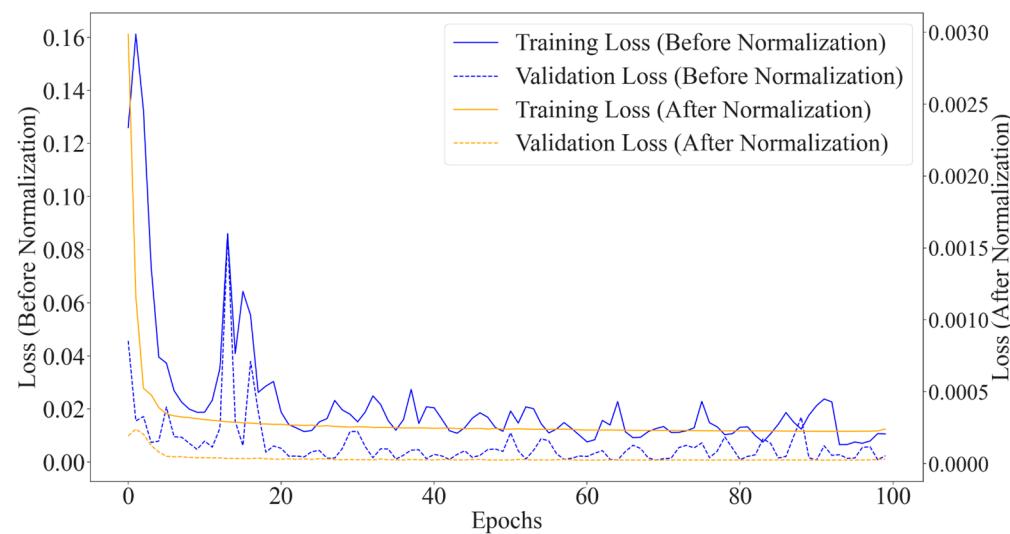


**Figure 3.** Test dataset and training dataset of GNSS time series.

Figure 3 illustrates the displacement in the North direction over time, comprising a total of about 4700 h. The training dataset exhibits noise and displacement over approximately 3700 h, whereas the test dataset displays fewer outliers over about 930 h. This partitioning strategy aims to assess the prediction accuracy of the proposed CNN-GRU method in the presence of noise and displacement. This figure serves as a visual representation of the dataset's temporal characteristics and the delineation of training and validation subsets, essential for robust model development and evaluation in time series analysis tasks.

In Figure 4, the change in the loss function during the training of the CNN-GRU model before and after data normalization is illustrated. Mean Squared Error (MSE) is

selected as the loss function, which is the most commonly used evaluation metric for neural networks in regression problems. As depicted in the figure, before data normalization, both the training and testing sets exhibit large loss function values, slow convergence, and post-convergence oscillations. However, after data normalization, the loss function values for both sets decrease by two orders of magnitude, with faster convergence and no post-convergence oscillations observed. Additionally, the consistency between the training and testing sets is improved after convergence. In summary, data normalization plays a crucial role in enhancing the performance of the CNN-GRU neural network.



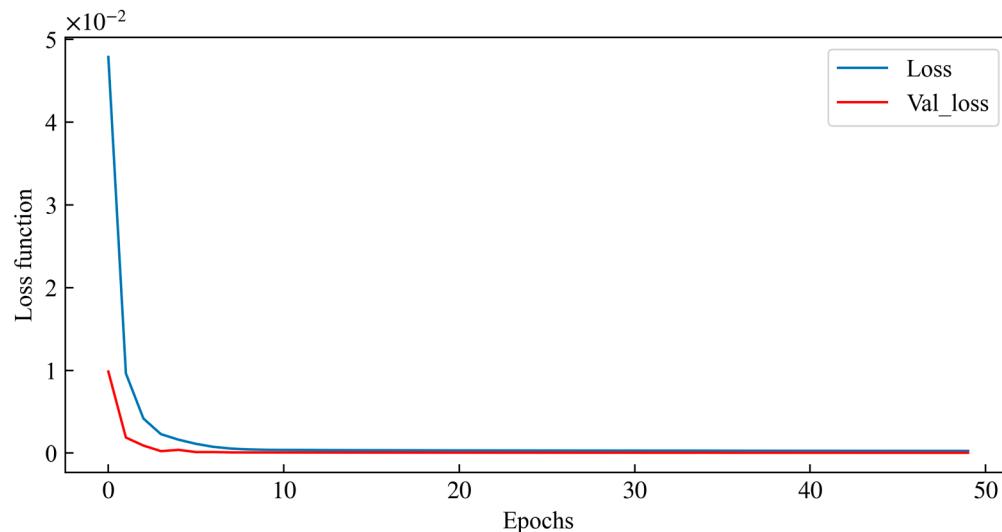
**Figure 4.** The change in the loss function before and after data normalization.

To validate the performance of the proposed CNN-GRU method, the experimental section of this paper is divided into three parts. The first part consists of experiments on individual CNN and GRU models, aimed at analyzing the shortcomings of individual models. The second part is the experimental verification of the CNN-GRU model proposed in this paper, highlighting the advantages of the proposed model. The third part involves parameter-tuning experiments for the CNN-GRU model.

### 3.1. The Experiment for the Dividual Model

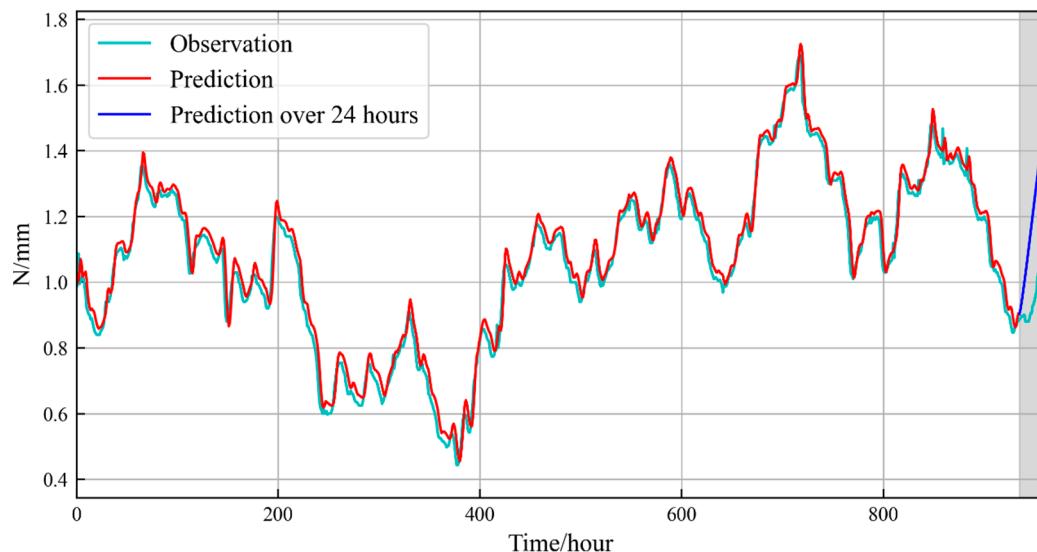
The first experiment aimed to analyze and validate the individual cases of the CNN model and GRU model. The CNN model and GRU model are conducted separately using the aforementioned data. The loss function of the CNN modeling is shown in the following Figure 5. The loss function is typically used to measure the difference between the model's predictions and observations, serving as the objective to be minimized during optimization. To validate the performance of the neural network, 50 training epochs were used for iterative training in the experiments. The loss function is the MSE parameter in Equation (8).

As for the CNN model, the change in the loss function graph is used to describe the variation of loss values during the model training process. During training, the change in the loss function graph reflects the trend of loss values across different epochs. In Figure 5, the loss function of the CNN sharply decreases from the 1st to the 5th epoch, and then gradually stabilizes from the 6th to the 10th iteration. Between the 10th and 50th epoch, the change in the loss function is extremely small, indicating that the model gradually fits the training dataset better. The CNN model typically requires approximately 10 training epochs to achieve consistency between the training dataset and the test dataset.



**Figure 5.** The loss function diagram of the CNN model.

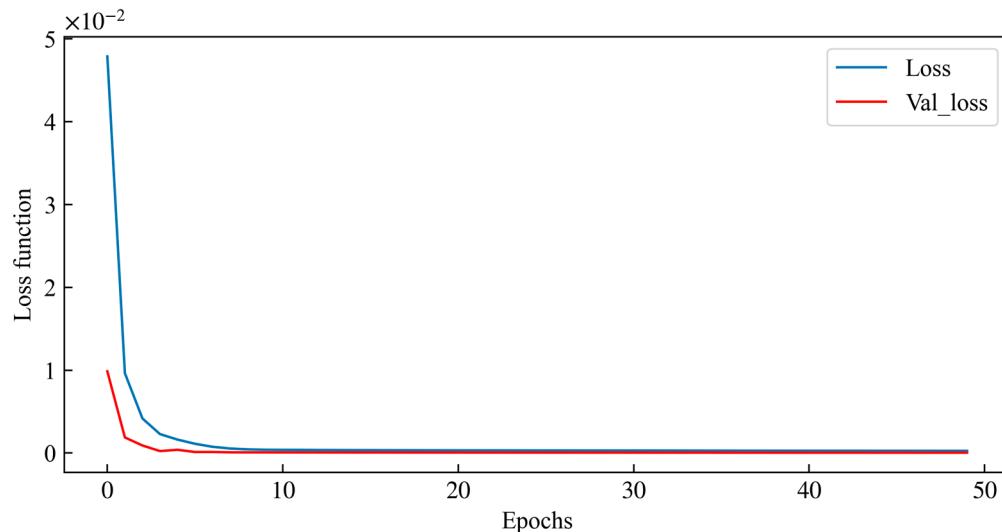
Figure 6 presents the variation curve of the observations and predictions in the test dataset and the extrapolated predictions over 24 h in the CNN model. Overall, there is good consistency between the observations and predictions in the test dataset, with fewer outliers and smoother curves in the predictions. In the interval around the 870th hour, the predictions have fewer and smaller abnormal fluctuations, demonstrating the superiority of the CNN model in GNSS time series modeling and predictions. However, the predictions have a systematic upward deviation relative to the observations, approximately 0.04 mm, which has a minimal impact on GNSS positioning. Additionally, the extrapolated predictions of the CNN model have the same trend as the observation, but there is a significant discrepancy between the predictions and observations in the extrapolated predictions over 24 h.



**Figure 6.** The time series diagram of the CNN model.

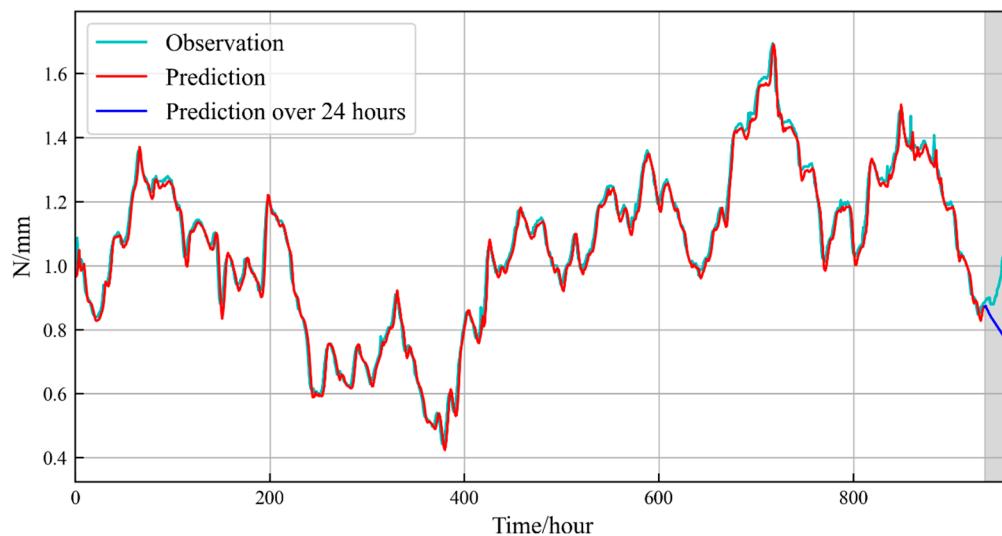
The above experiment shows the strengths and weaknesses of the CNN model. The change in Figure 7 is used to describe the variation in the loss values of the GRU model. The loss function of the GRU sharply decreases from the 1st to the 4th epoch, and then gradually stabilizes from the 5th to the 12th iteration. Between the 13th and 50th epoch, the change in the loss function is extremely small, indicating that the model gradually fits

the training dataset better. The GRU model typically requires approximately 12 training epochs to achieve consistency between the training dataset and the test dataset.



**Figure 7.** The loss function diagram of the GRU model.

Similar to Figure 5, Figure 8 presents the variation curve of the observations and predictions of the GRU model. The predictions have good consistency with the observations in the test dataset. However, there are still some areas of obvious inconsistency between the two, such as around the 80th hour and the 700th hour. In the interval around the 870th hour, the predictions have fewer and smaller outliers of about 0.08 mm, demonstrating the superiority of the GRU model in GNSS time series modeling and predictions. However, in the extrapolated predictions over 24 h, there is a significant discrepancy between the predictions and observations, with a maximum error of approximately 0.2 mm. This suggests that while the GRU model can be applied to time series predictions in GNSS deformation monitoring, there are still some shortcomings in extrapolation predictions.



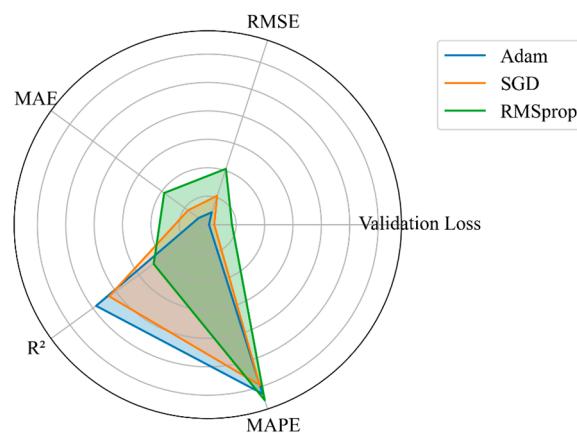
**Figure 8.** The time series diagram of the GRU model.

### 3.2. The Experiment for the CNN-GRU Model

The experiment in Section 3.1 analyzes the advantages and disadvantages of the CNN model and the GRU model separately. And this section will analyze the proposed CNN-GRU model. For comparison, the CNN-GRU model uses the same parameters as the CNN

and GRU models. The above experiment selects the adaptive moment estimation (Adam) optimizer for the CNN and GRU models. This section will analyze the performance of different optimizers of the proposed CNN-GRU model.

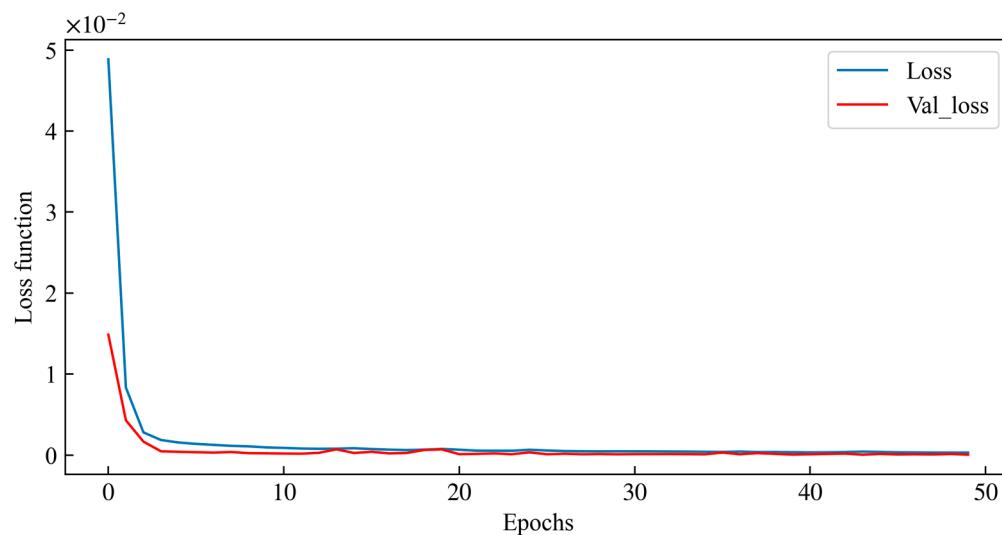
The comparative evaluation of Adam, stochastic gradient descent (SGD), and Root Mean Square Propagation (RMSProp) optimizers are presented in Figure 9. The radar diagram, including the mean absolute percentage error (MAPE), root mean square error (RMSE), mean absolute error (MAE),  $R^2$ , and validation loss, provides a comprehensive analysis of the performance of different optimizers. The average values from 10 training epochs are used as evaluation metrics. It is evident that the Adam optimizer has the best fitting with the largest  $R^2$  and the smallest MAE, RMSE, and Validation Loss. The SGD optimizer has a slightly inferior performance than the Adam optimizer. And the RMSProp optimizer shows the worst-fitting performance. As for the MAPE, the differences among the three optimizers are negligible. In general, the Adam optimizer has the best performance for the prediction of GNSS deformation monitoring. The subsequent sections will take the Adam optimizer as an example for the CNN-GRU model.



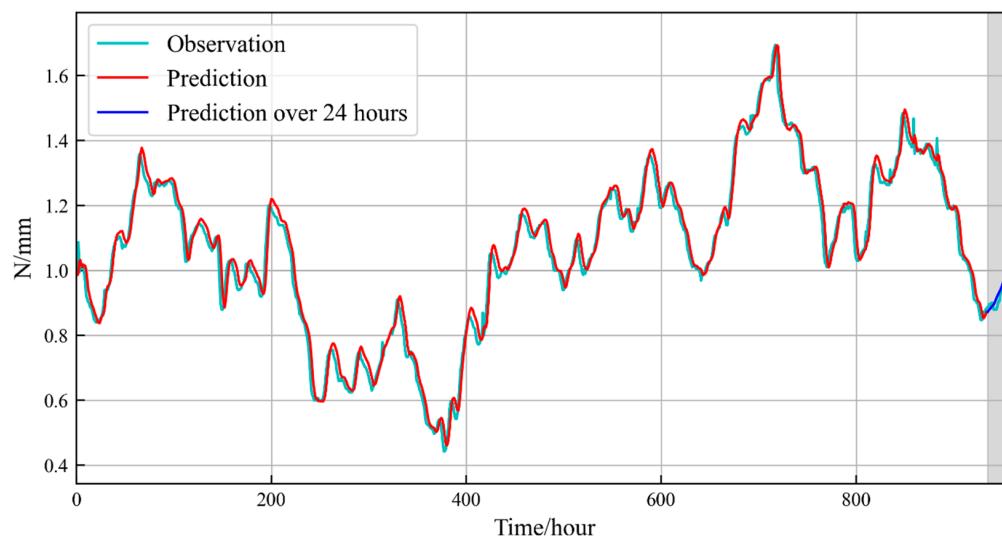
**Figure 9.** The Radar diagram for comparing different optimizers.

The above experiment shows the strengths and weaknesses of the CNN model and the GRU model. The change in Figure 10 is used to describe the variation of loss values of the proposed CNN-GRU model. The loss function of the CNN-GRU model sharply decreases from the 1st to the 4th epoch, and then gradually stabilizes from the 5th to the 21st iteration. Between the 22nd and 50th epoch, the change in the loss function is extremely small, indicating that the model gradually fits the training dataset better. The GRU model typically requires approximately 22 training epochs to achieve consistency between the training dataset and the test dataset.

Figure 11 presents the variation curve of the observations and predictions of the proposed CNN-GRU model. The predictions have good consistency with the observations in the test dataset. In contrast with Figures 6 and 8, the predictions in Figure 11 have a better performance of consistency and smoothness than the CNN model and the GRU model. However, there are still some areas of obvious inconsistency between the two, such as around the 200th hour and around the 820th hour. In the interval around the 870th hour, the predictions have fewer and smaller outliers. Additionally, the extrapolated predictions of the CNN-GRU model are better than the CNN and GRU model, which is almost consistent with the observations. Figure 11 demonstrates the superiority of the proposed CNN-GRU model in GNSS time series modeling and predictions.



**Figure 10.** The loss function diagram of the CNN-GRU model.

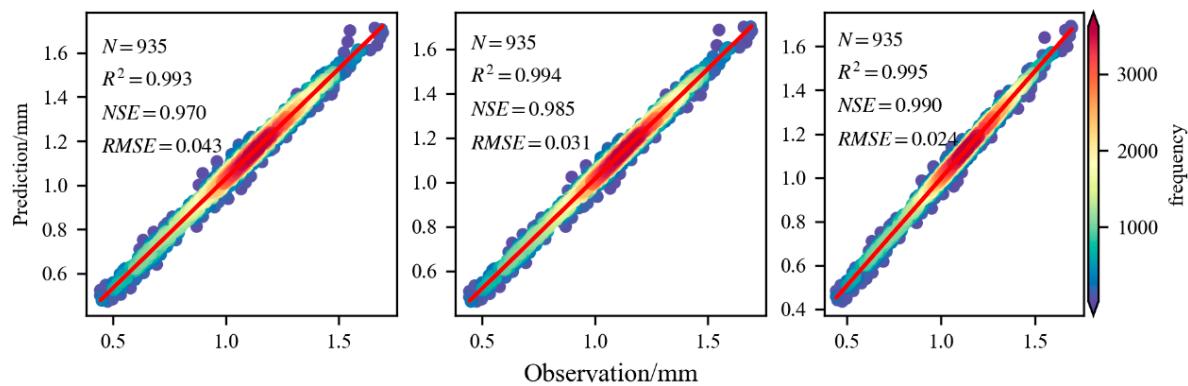


**Figure 11.** The time series diagram of the CNN-GRU model.

### 3.3. The Experiment for Comparison

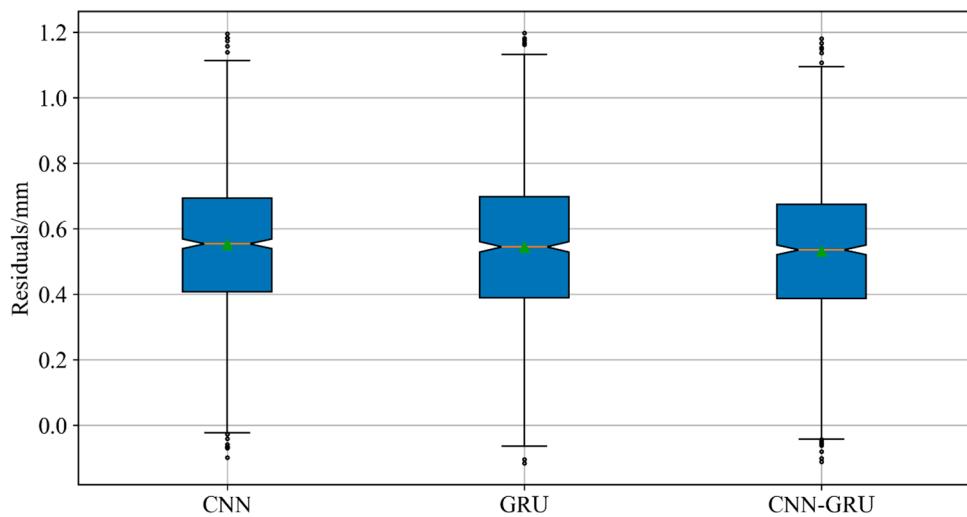
The above experiments provide a visual comparison of the CNN model, the GRU model, and the proposed CNN-GRU model. More details will be provided in the following experiment.

Figures 6, 8 and 11 indicate that these three models have good consistency with the actual observations. Figure 12 has a similar conclusion, with the CNN-GRU model performing the best, followed by the GRU model, and the CNN model performing the worst. In terms of the  $R^2$  comparison, the CNN-GRU model has the best performance, with little difference from the other two models. NSE (Nash–Sutcliffe Efficiency) is a metric used to evaluate the matching degree between predictions and observations in hydrological models, which is used to assess the consistency between predictions and observations in GNSS deformation monitoring for hydraulic structures. The CNN-GRU model has the best performance, while the CNN model has the worst performance. Similar to NSE, the CNN-GRU model significantly outperforms the CNN and GRU models in terms of the RMSE. In addition, the proposed method has an improvement of about 45% compared with the CNN model. In summary, the proposed CNN-GRU model combines the advantages of CNN and GRU, resulting in a better predictive performance.



**Figure 12.** The comparison of the heatmap (the left is the CNN model, the middle is the GRU model, and the right is the CNN-GRU model).

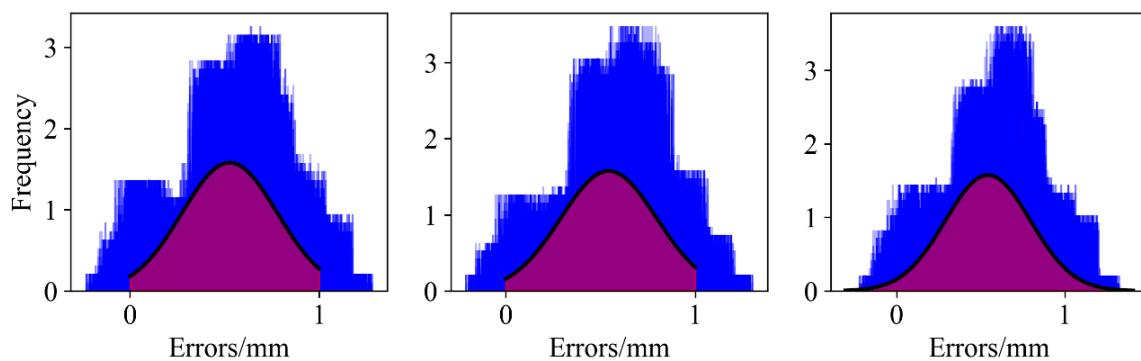
Figure 13 shows the comparison of the residual between the three models. The CNN model has the worst performance, with a maximum residual of about 1.2 mm. The GRU model has a better performance, with a maximum residual of about 1.2 mm. And it has fewer outliers. Furthermore, the proposed CNN-GRU model has the best performance, with a maximum residual of about 1.19 mm. And it has smaller median residuals. This proves that the proposed CNN-GRU model has a better applicability in GNSS deformation monitoring for hydraulic structures.



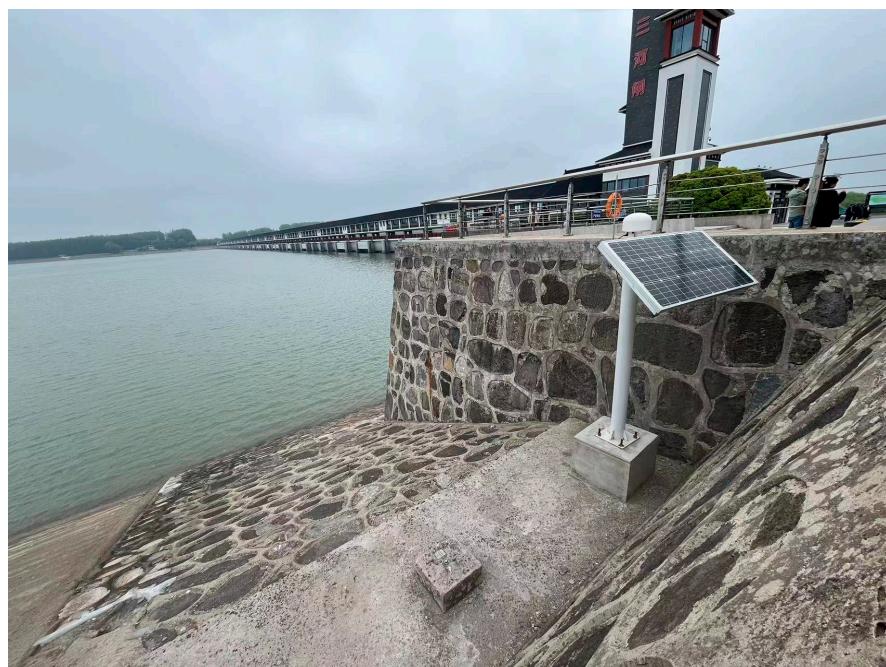
**Figure 13.** Box diagram of the residuals.

For further comparison, Figure 14 compares the error statistics of the CNN model, the GRU model method, and the proposed CNN-GRU model. The left panel displays the error distribution and curve fitting of the CNN model, with errors mainly concentrated in the  $(-0.4, 1.4)$  range and a central value of approximately 0.7 mm for the fitted curve. The middle is for the GRU model, with errors mainly concentrated in the  $(-0.3, 1.4)$  range and a central value of approximately 0.7 mm for the fitted curve. The right is for the CNN-GRU model, with errors primarily distributed in the  $(-0.2, 1.2)$  range and a central value of about 0.7 mm for the fitted curve. In general, the proposed method, by combining the CNN model and the GRU model, achieves a better prediction performance.

The above experiment demonstrates the feasibility of the proposed CNN-GRU method. This section presents a comparison between the proposed method and the Extended Kalman Filter (EKF). The experiment was also conducted at Sanhe Sluice, with another monitoring point illustrated in Figure 15. The experimental period was from 10 September 2023 to 24 April 2024 and other configure parameters are the same as the above experiment.



**Figure 14.** The error histogram (the left is the CNN model, the middle is the GRU model, and the right is the CNN-GRU model).



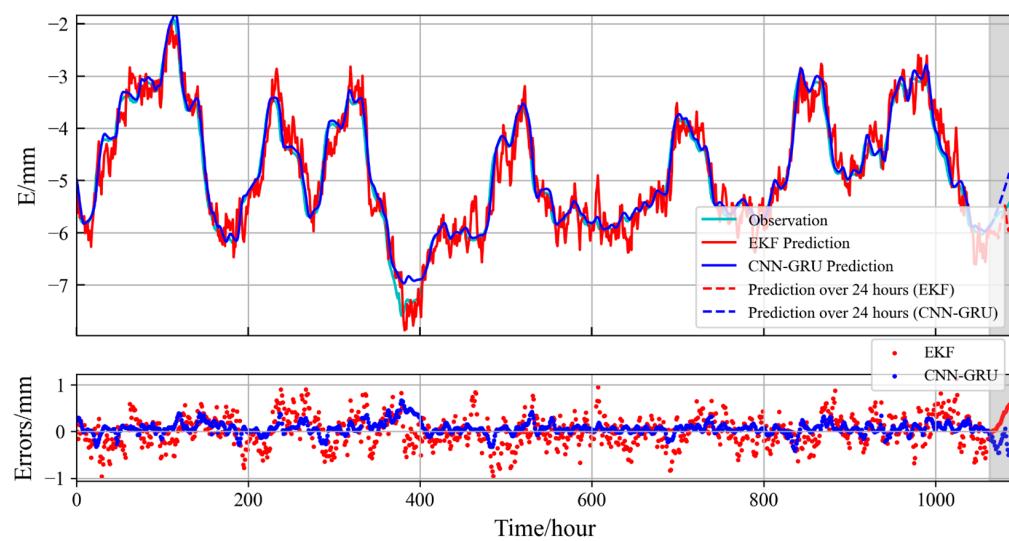
**Figure 15.** Another monitoring site used for comparative experiment.

Table 2 illustrates the performance metrics for both the EKF and the proposed CNN-GRU method. A comprehensive evaluation indicates significant improvements across all metrics with the CNN-GRU method compared to the EKF, showcasing its superior accuracy and reliability in GNSS deformation monitoring for hydraulic structures. Specifically, in comparison to the EKF, the adoption of the CNN-GRU method results in a remarkable 92.41% reduction in the MSE, a notable 72.44% decrease in the RMSE, a substantial 76.29% decrease in the MAE, and a noteworthy 8.02% increase in the coefficient of the determination of  $R^2$ . In summary, the CNN-GRU method demonstrates clear advantages in geophysical process analysis, offering more precise and dependable outcomes for geological structure analysis.

**Table 2.** The comparison of the CNN-GRU and EKF.

	MSE (mm)	RMSE (mm)	MAE (mm)	$R^2$
EKF	0.1646	0.4057	0.3236	0.9037
CNN-GRU	0.0125	0.1118	0.0767	0.9920

Figure 16 illustrates a comparative analysis between the CNN-GRU and EKF methodologies, focusing on both the test dataset and a 24 h extrapolation scenario. The figure comprises a time series plot and an error scatter plot. In the time series plot, the CNN-GRU method demonstrates closer adherence to the original observations from the test dataset, while the EKF method exhibits more frequent fluctuations throughout the process. This observation is further supported by the error scatter plot, which indicates the superior performance of the CNN-GRU approach over EKF. Specifically, around the 240th hour, EKF presents prediction errors of approximately 1 mm, whereas CNN-GRU maintains a more consistent behavior with fewer and less pronounced deviations over the entire interval, reaching a maximum of about 0.8 mm. Overall, the CNN-GRU methodology outperforms EKF, delivering predictions characterized by higher accuracy and stability.



**Figure 16.** The prediction diagram of the CNN-GRU and EKF method.

#### 4. Discussion

Compared to the traditional EKF method, CNN-GRU demonstrates significant advantages in GNSS deformation monitoring. These are primarily manifested in its superior capability for nonlinear modeling, stronger spatiotemporal correlation, enhanced data representation and learning, as well as higher prediction accuracy and reliability. By effectively capturing the nonlinear relationships and spatiotemporal correlations within the data, the CNN-GRU method can provide more precise and reliable predictions for deformation monitoring, thus ensuring greater safety and stability in hydraulic engineering projects. The experiment shows that the CNN-GRU method has about 76% improvement against the EKF method.

The CNN model extracts features from the GNSS time series by stacking convolutional layers, pooling layers, and fully connected layers, gradually combining these features for prediction. It has strong feature extraction capabilities and translation invariance. However, it remains sensitive to changes in the scale and position of the target and requires large amounts of data. This study carries on experiments with only the ReLU activation function and the Adam optimizer. In the future, more activation functions and optimizers will be tested to accelerate the model convergence and improve its performance.

The GRU model improves the training effectiveness by introducing gate mechanisms to address issues like gradient vanishing and exploding in traditional RNN models. However, its ability to model long-term dependencies is limited, and it does not fully exploit the information in the middle part of the GNSS time series. Introducing attention mechanisms in the future will enable the model to better focus on important parts of the GNSS time series, enhancing its ability to handle information in the middle of the GNSS time series.

The proposed CNN-GRU model utilizes the CNN model to extract features from the GNSS time series and the GRU model to capture long-term dependencies, achieving better performances in modeling. It implements hierarchical feature learning, effectively extracting features from GNSS time series and capturing long-term dependencies, thus enhancing the model's ability for GNSS time series. The proposed method has four advantages:

- (1) The proposed method comprehensively utilizes spatial and temporal features, enabling a more thorough analysis of the data;
- (2) The proposed method reduces information loss and enhances model performance;
- (3) The proposed method can more accurately capture spatial and temporal features of the data, thereby improving prediction accuracy;
- (4) The proposed method combines the advantages of both models, enabling better balance between model complexity and generalization ability, thus reducing the risk of overfitting.

However, parameter tuning is complex, and training is time-consuming. In the future, structural optimization and attention mechanisms will be introduced to enable the model to better focus on important parts of the GNSS time series, improving model performance.

## 5. Conclusions

In this study, we proposed a novel application of the CNN-GRU model for GNSS deformation monitoring. Our objective is to leverage the capabilities of deep learning in capturing complex spatiotemporal patterns inherent in GNSS time series to enhance modeling and prediction in hydraulic structures. Through comprehensive analysis and experimentation, we demonstrated the effectiveness of the CNN-GRU model in modeling and prediction for GNSS time series. The CNN model efficiently extracted features, capturing localized deformations and spatial dependencies. Meanwhile, the GRU model effectively models the temporal dynamics and long-term dependencies, enabling the accurate prediction of future deformations.

Our results indicated that the CNN-GRU model outperformed traditional methods with about 45% improvement. By jointly leveraging convolutional and recurrent architectures, the proposed model achieved superior performance in capturing both spatial and temporal characteristics of deformation patterns, thereby enhancing the accuracy and reliability of deformation prediction in hydraulic applications.

Overall, this study demonstrates the potential of deep learning approaches, particularly the CNN-GRU model, in advancing GNSS deformation monitoring and prediction capabilities in GNSS deformation monitoring for hydraulic structures. By effectively integrating spatial and temporal information, this approach offers a promising avenue for improving the efficiency, accuracy, and reliability of deformation monitoring systems, ultimately contributing to the safety and sustainability of a hydraulic infrastructure.

**Author Contributions:** Conceptualization, Y.X., J.W. and A.D.; Data curation, H.L. and Y.K.; Formal analysis, Y.X. and H.L.; Funding acquisition, A.D. and Y.X.; Methodology, H.L., Y.K. and Y.X.; Project administration, J.W.; Resources, J.Z. and Y.W.; Software, H.L. and Y.Y.; Writing—original draft, H.L.; Writing—review and editing, J.Z. and Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Jiangsu Provincial Department of Water Resources, grant number 2022050 and 2022083, the Jiangsu Hydraulic Research Institute, grant number 2023z036 and 2023z045, and the Department of Natural Resources of the Hunan Province, grant number 20230104.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Delzon, S.; Cochard, H. Recent advances in tree hydraulics highlight the ecological significance of the hydraulic safety margin. *New Phytol.* **2014**, *203*, 355–358. [[CrossRef](#)] [[PubMed](#)]
2. Grossiord, C.; Ulrich, D.E.; Vilagrosa, A. Controls of the hydraulic safety–efficiency trade-off. *Tree Physiol.* **2020**, *40*, 573–576. [[CrossRef](#)]
3. Sperry, J.S.; Meinzer, F.C.; McCULLOH, K.A. Safety and efficiency conflicts in hydraulic architecture: Scaling from tissues to trees. *Plant Cell Environ.* **2008**, *31*, 632–645. [[CrossRef](#)]
4. De Vriend, H.J.; van Koningsveld, M.; Aarninkhof, S.G.; de Vries, M.B.; Baptist, M.J. Sustainable hydraulic engineering through building with nature. *J. Hydro-Environ. Res.* **2015**, *9*, 159–171. [[CrossRef](#)]
5. Parabuchev, I. Problems of and ways to optimize engineering surveys and investigations for modern hydraulic construction. *Hydrotech. Constr.* **2000**, *34*, 145–149. [[CrossRef](#)]
6. Li, S.; Wang, X.; Xu, Z.; Mao, D.; Pan, D. Numerical investigation of hydraulic tomography for mapping karst conduits and its connectivity. *Eng. Geol.* **2021**, *281*, 105967. [[CrossRef](#)]
7. Zha, Y.; Yeh, T.-C.J.; Illman, W.A.; Tanaka, T.; Bruines, P.; Onoe, H.; Saegusa, H. What does hydraulic tomography tell us about fractured geological media? A field study and synthetic experiments. *J. Hydrol.* **2015**, *531*, 17–30. [[CrossRef](#)]
8. Tretyak, K.; Palianytsia, B. Dam Spatial Temperature Deformations Model Development Based on GNSS Data. *J. Perform. Constr. Facil.* **2023**, *37*, 04023028. [[CrossRef](#)]
9. Zhang, P.; Pang, Z.; Lu, J.; Jiang, W.; Sun, M. Real-Time Water Level Monitoring Based on GNSS Dual-Antenna Attitude Measurement. *Remote Sens.* **2023**, *15*, 3119. [[CrossRef](#)]
10. Sabeti, H.; Pourmina, A.; Rezaei, A.; Tayfehrostami, A.; Einlou, F.; Nankali, H. Discovering confined zones and land deformation characteristics across an aquifer system in Iran using GNSS and InSAR techniques. *Hydrogeol. J.* **2023**, *31*, 2061–2076. [[CrossRef](#)]
11. Lu, C.; Zhong, Y.; Wu, Z.; Zheng, Y.; Wang, Q. A tropospheric delay model to integrate ERA5 and GNSS reference network for mountainous areas: Application to precise point positioning. *GPS Solut.* **2023**, *27*, 81. [[CrossRef](#)]
12. Shu, B.; He, Y.; Wang, L.; Zhang, Q.; Li, X.; Qu, X.; Huang, G.; Qu, W. Real-time high-precision landslide displacement monitoring based on a GNSS CORS network. *Measurement* **2023**, *217*, 113056. [[CrossRef](#)]
13. Nzelibe, I.U.; Tata, H.; Idowu, T.O. Assessment of GNSS zenith tropospheric delay responses to atmospheric variables derived from ERA5 data over Nigeria. *Satell. Navig.* **2023**, *4*, 15. [[CrossRef](#)]
14. Weng, D.; Hou, Z.; Meng, Y.; Cai, M.; Chan, Y. Characterization and mitigation of urban GNSS multipath effects on smartphones. *Measurement* **2023**, *223*, 113766. [[CrossRef](#)]
15. Yuan, H.; Zhang, Z.; He, X.; Dong, Y.; Zeng, J.; Li, B. Multipath mitigation in GNSS precise point positioning using multipath hierarchy for changing environments. *GPS Solut.* **2023**, *27*, 193. [[CrossRef](#)]
16. Xu, P.; Zhang, G.; Zhong, Y.; Yang, B.; Hsu, L.-T. A Framework for Graphical GNSS Multipath and NLOS Mitigation. *IEEE Trans. Intell. Transp. Syst.* **2024**, *1*–11. [[CrossRef](#)]
17. Zhang, Q.; Zhang, L.; Sun, A.; Meng, X.; Zhao, D.; Hancock, C. GNSS Carrier-Phase Multipath Modeling and Correction: A Review and Prospect of Data Processing Methods. *Remote Sens.* **2024**, *16*, 189. [[CrossRef](#)]
18. Zhou, T.; Jie, Y.; Wei, Y.; Zhang, Y.; Chen, H. A real-time prediction interval correction method with an unscented Kalman filter for settlement monitoring of a power station dam. *Sci. Rep.* **2023**, *13*, 4055. [[CrossRef](#)] [[PubMed](#)]
19. Yin, Z.; Yang, J.; Ma, Y.; Wang, S.; Chai, D.; Cui, H. A Robust Adaptive Extended Kalman Filter Based on an Improved Measurement Noise Covariance Matrix for the Monitoring and Isolation of Abnormal Disturbances in GNSS/INS Vehicle Navigation. *Remote Sens.* **2023**, *15*, 4125. [[CrossRef](#)]
20. Roux, A.; Changey, S.; Lauffenburger, J.-P.; Weber, J. Mortar Trajectory Estimation by a Deep Error-State Kalman Filter in a GNSS-Denied Environment. In Proceedings of the 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 24–27 April 2023; pp. 119–127.
21. Liu, J.; Achurra, A.; Zhang, C.; Bury, A.; Wang, X. A long short term memory network-based, global navigation satellite system/inertial navigation system for unmanned surface vessels. *J. Mar. Eng. Technol.* **2024**, *1*–13. [[CrossRef](#)]
22. Tasan, M.; Ghorbaninasab, Z.; Haji-Aghajany, S.; Ghiasvand, A. Leveraging GNSS tropospheric products for machine learning-based land subsidence prediction. *Earth Sci. Inform.* **2023**, *16*, 3039–3056. [[CrossRef](#)]
23. Becker, J.M.; Roggenbuck, O. Prediction of Significant Wave Heights with Engineered Features from GNSS Reflectometry. *Remote Sens.* **2023**, *15*, 822. [[CrossRef](#)]
24. Shangguan, M.; Dang, M.; Yue, Y.; Zou, R. A Combined model to predict GNSS precipitable water vapor based on deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 4713–4723. [[CrossRef](#)]
25. Ng, K.M.; Nahar, R.S.; IbneReaz, M. Linear regression models with autoregressive integrated moving average errors for measurements from real time kinematics-global navigation satellite system during dynamic test. *Int. J. Electr. Comput. Eng. IJECE* **2023**, *13*, 770–780. [[CrossRef](#)]
26. Shen, L.; Lv, W. Mine Surface Settlement Prediction Based on Optimized VMD and Multi-Model Combination. *Processes* **2023**, *11*, 3309. [[CrossRef](#)]
27. Li, Z.; Lu, T.; Yu, K.; Wang, J. Interpolation of GNSS Position Time Series Using GBDT, XGBoost, and RF Machine Learning Algorithms and Models Error Analysis. *Remote Sens.* **2023**, *15*, 4374. [[CrossRef](#)]

28. Jiang, W.; Wang, J.; Li, Z.; Li, W.; Yuan, P. A new deep self-attention neural network for GNSS coordinate time series prediction. *GPS Solut.* **2024**, *28*, 3. [[CrossRef](#)]
29. Chen, H.; Lu, T.; Huang, J.; He, X.; Yu, K.; Sun, X.; Ma, X.; Huang, Z. An Improved VMD-LSTM Model for Time-Varying GNSS Time Series Prediction with Temporally Correlated Noise. *Remote Sens.* **2023**, *15*, 3694. [[CrossRef](#)]
30. Petrovski, I.G. RTK-B Enhancing Real-Time Kinematic (RTK) Positioning with Genetic Algorithms and Long-Code GNSS Signals for Rapid Ambiguity Resolution and Improved Accuracy. In *International Workshop on Advances in Civil Aviation Systems Development*; Springer: Cham, Switzerland, 2023; pp. 116–125.
31. Jamali, S.E.; Masud, M.M. Orbit Propagation and Determination using Genetic Algorithms. In Proceedings of the 2023 15th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 14–15 November 2023; pp. 13–19.
32. Han, Y.; Wang, L.; Chen, R.; Fu, W.; Li, T.; Liu, G.; Zhou, H. Topside ionospheric TEC modeling using multiple LEO satellites based on genetic algorithm-optimized machine learning models. *GPS Solut.* **2024**, *28*, 19. [[CrossRef](#)]
33. De la Fraga, L.G.; Ovilla-Martínez, B.; Tlelo-Cuautle, E. Echo state network implementation for chaotic time series prediction. *Micropress. Microsyst.* **2023**, *103*, 104950. [[CrossRef](#)]
34. González-Zapata, A.M.; Tlelo-Cuautle, E.; Cruz-Vega, I. On the optimization of machine learning techniques for chaotic time series prediction. *Electronics* **2022**, *11*, 3612. [[CrossRef](#)]
35. Tang, J.; Liu, C.; Yang, D.; Ding, M. Prediction of ionospheric TEC using a GRU mechanism method. *Adv. Space Res.* **2024**. [[CrossRef](#)]
36. Xu, P.; Zhang, G.; Yang, B.; Hsu, L.-T. PositionNet: CNN-based GNSS positioning in urban areas with residual maps. *Appl. Soft Comput.* **2023**, *148*, 110882. [[CrossRef](#)]
37. Mehr, I.A.E.; Dovis, F. A deep neural network approach for detection and classification of GNSS interference and jammer. *TechRxiv* **2023**. [[CrossRef](#)]
38. Meng, Y.; Qin, Y.; Cai, Z.; Tian, B.; Yuan, C.; Zhang, X.; Zuo, Q. Dynamic forecast model for landslide displacement with step-like deformation by applying GRU with EMD and error correction. *Bull. Eng. Geol. Environ.* **2023**, *82*, 211. [[CrossRef](#)]
39. Tong, R.; Liu, C.; Tao, Y.; Wang, X.; Sun, J. ConvGRU-MHM: A CNN GRU-enhanced MHM for mitigating GNSS multipath. *Meas. Sci. Technol.* **2024**, *35*, 045007. [[CrossRef](#)]
40. Zhao, Z.; Yun, S.; Jia, L.; Guo, J.; Meng, Y.; He, N.; Li, X.; Shi, J.; Yang, L. Hybrid VMD-CNN-GRU-based model for short-term forecasting of wind power considering spatio-temporal features. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105982. [[CrossRef](#)]
41. Guo, Z.; Yang, C.; Wang, D.; Liu, H. A novel deep learning model integrating CNN and GRU to predict particulate matter concentrations. *Process Saf. Environ. Prot.* **2023**, *173*, 604–613. [[CrossRef](#)]
42. Hua, G.; Wang, S.; Xiao, M.; Hu, S. Research on the uplift pressure prediction of concrete dams based on the CNN-GRU model. *Water* **2023**, *15*, 319. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.