electronics

MDPI

Article

Short-Term Flood Prediction Model Based on Pre-Training Enhancement

Yang Xia * and Jiamin Lu

College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China
* Correspondence: 211307040023@hhu.edu.cn

Abstract: With the rapid advancement of deep learning techniques, deep learning-based flood prediction models have drawn significant attention. However, for short-term prediction in small- and medium-sized river basins, models typically rely on hydrological data spanning from the past several hours to one day, utilizing a fixed-length input window. Such input limits the models' adaptability to diverse rainfall events and restricts their capability to comprehend historical temporal patterns. To address the underutilization of historical information by existing models, we introduce a Pre-training Enhanced Short-term Flood Prediction Method (PE-SFPM) to enrich the model's temporal understanding without necessitating changes to the input window size. In the pre-training stage, the model uses a random masking and prediction strategy to learn segment features, capturing a more comprehensive evolutionary trend of historical floods. In the flow forecasting stage, temporal features and spatial features are captured and fused using the temporal attention, spatial attention, and gated fusion. Features are further enhanced by integrating segment features using a feed-forward network. Experimental results demonstrate that the proposed PE-SFPM model achieves excellent performance in short-term flood prediction tasks.

Keywords: Artificial Intelligence; short-term flood prediction; pre-training

check for updates

Citation: Xia, Y.; Lu, J. Short-Term Flood Prediction Model Based on Pre-Training Enhancement. *Electronics* **2024**, *13*, 2203. https://doi.org/10.3390/electronics13112203

Academic Editors: Faheim Sufi, Alam Edris and Aryya Gangopadhyay

Received: 6 May 2024
Revised: 31 May 2024
Accepted: 4 June 2024
Published: 5 June 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

Flood disasters, as one of the most catastrophic natural phenomena, have had immense impacts on economic and social development. Effective flood prediction methods are thus of paramount importance for emergency disaster response and mitigation [1]. By analyzing hydrological data, flood prediction models can forecast river flow changes over the coming hours to days, offering crucial support for disaster mitigation efforts. In recent years, the rise of deep learning technologies has revolutionized flood prediction and management research [2], introducing innovative approaches to address various challenges [3–13]. This paper focuses primarily on short-term flood prediction for small and medium-sized basins.

Although many models [6–13] have achieved good results, they still have limitations. When making short-term flood forecasting for small- and medium-sized basins, these models often use hydrological data from a few hours up to one day with a fixed input window. Due to the complexity and variability of flood events, these models may struggle to forecast varying future trends using fixed short-term input data, and while extending the range of historical data input could theoretically enhance prediction accuracy, it often introduces noise, increases model training complexity, and raises computational demands [12,13], resulting in minimal performance gains. Therefore, researchers face the challenge of improving the model's sensitivity to historical data without altering the input window size.

Drawing inspiration from pre-training models such as BERT [14] and MAE [15], we introduce the Pre-training Enhanced Short-term Flood Prediction Model (PE-SFPM). This innovative approach aims to sharpen the model's historical data perception via pre-training without altering the input window. PE-SFPM is explicitly crafted to discern and learn

significant temporal patterns during its pre-training stage, and to make flood predictions utilizing convolution, attention, and gated fusion in the flow forecasting stage.

In the pre-training stage, hydrological data undergo segmentation and are subjected to convolution and attention with a random masking and prediction strategy. This technique is formulated to unearth and embody temporal patterns to encompass insights from extended historical timelines and crystallize segment features reflective of complex hydrological scenarios. Advancing to the flow forecasting stage, PE-SFPM applies temporal and spatial attention to distill impactful data from discrete temporal junctures and varying station locales, creating intricate temporal and spatial features. These features are subsequently integrated via gated fusion. The resultant segment features serve as amplifiers for the features processed during the flow forecasting stage.

The efficacy and robustness of PE-SFPM are corroborated through meticulous comparative experiments conducted across diverse typical basin datasets. These studies validate the model's superiority in excelling established deep learning benchmarks.

In summary, the primary contributions are as follows:

- We propose a novel model, the Pre-training Enhanced Short-term Flood Prediction Model (PE-SFPM), specifically designed for short-term flood prediction in small- and medium-sized basins. The PE-SFPM utilizes pre-training to generate segment features that capture complex temporal patterns from historical data, enhancing the model's predictive capabilities without necessitating adjustment to the input window.
- We design a pre-training model that leverages convolution, relative position encoding, and masked attention mechanisms. In the pre-training stage, the model employs random masking and prediction to effectively learn temporal patterns. In the flow forecasting stage, the model uses dilated causal convolution, temporal attention, spatial attention, and gated fusion to extract and integrate temporal and spatial features, improving prediction accuracy.
- We conduct comparative experiments on three real-world basin datasets to validate the effectiveness of the proposed method. The experimental results demonstrate that the PE-SFPM model outperforms existing deep learning models in terms of prediction precision and robustness, showcasing its potential for practical applications in flood forecasting.

2. Background and Existing Literature

Flood prediction and management have seen significant advancements with the advent of deep learning technologies. This section reviews and summarizes existing literature and highlights the research problem addressed in this study.

UFM [3] allow information on river construction and underlying surface changes to be introduced into the flood hygrograph implicitly, thus avoiding the difficulty of establishing hydrological and hydrodynamic coupling of the whole basin. RRFFM [4] mitigates the impact of outliers using a robust loss function while introducing a fluctuation coefficient to capture the volatile inflows into reservoirs. FDPTRC [5] constructs a three-dimensional Computational Fluid Dynamics (CFD) model using FLUENT and examines the effects of crucial roadway design features under varying flow rates and boundary conditions. STA-LSTM (2020) [6] utilizes dynamic attention mechanisms and LSTM, demonstrating the approach's validity through the visualization and interpretation of spatial and temporal attention weights. ST-GCN [7] utilizes GCN and LSTM to individually capture spatial and temporal features, simulating spatio-temporal relations, and introduces a temporal attention mechanism to assess the importance of different time steps, thereby enhancing flood prediction accuracy. The Attention-LSTM [8] simulates the long-term relationship between historical flood factors using LSTM, ensuring forecast precision by focusing on future rainfall trends and initial conditions such as early precipitation and river levels. AGCLSTM [9] leverages spatio-temporal GCN modules to grasp spatio-temporal features and spatio-temporal LSTM modules to extract dynamic spatio-temporal correlations. Experimentation confirms the model's superior performance in predicting flood peaks and

calibrating flow rates. DAGAT [10] uses the distribution adaptation mechanism of the Boosting algorithm to train weight parameters, enabling the reduction of distribution differences among different segmented periods and effectively improving the accuracy of flood forecasting. STA-LSTM (2023) [11] uses attention to focus on the more valuable feature factors in time and space, thus assign scores to input sequences. DSTA [12] uses spatial attention to capture the spatial correlation of each feature and then uses the temporal attention to select the corresponding time step for stream-flow series prediction. stResNet-LUBE [13] optimizes the coverage width-based criterion (CWC) objective function based on LUBE to adapt to the hydrological field and uses ResNet to avoid gradient disappearance in deep networks. The summary of the existing literature is shown in Table 1.

Table 1. Summary of existing flood prediction models.

Model	Applicable Scenarios	Technology Used	Performance	Disadvantages
UFM [3]	Urban flood risk prediction management	The solution of 3D Reynolds averaged NS equations	Effective for assessing flood risk based on different return periods	Only considers the flood peak discharge
RRFFM [4]	Flood forecasting for reservoirs	Robust error estimation, fluctuation coefficient, Xinanjiang model	Efficient and stable in the presence of data outliers	Only utilized in conjunction with the Xinanjiang model
FDPTRC [5]	Flow distribution pattern at T-shape roadway crossing	FLUENT, 3D CFD model	Provides accurate predictions for T-shape road crossings	Did not cover the impact of different roadway widths
STA-LSTM (2020) [6]	Flood prediction	LSTM, attention, Adam algorithm	Effective attention weight visualization	The graph information of basin is not used
ST-GCN [7]	Flood prediction	GCN, LSTM, attention	Effective for varying prediction horizons	The difference in hydrological station weights was not considered.
Attention-LSTM [8]	Flood prediction for small- and medium-sized river basins	LSTM, attention	Outperforms traditional ML models and LSTM	The spatial information in the basin was not considered.
AGCLSTM [9]	Flood prediction for small- and medium-sized river basins	GCN, LSTM, attention	Superior in flood prediction and flow calibration	Complex factors not fully explored
DAGAT [10]	Flood forecasting	GAT, GRU, Boosting	Reduces distribution different segmented periods	Explanations of the distribution and changes in the dataset needed.
STA-LSTM (2023) [11]	Water level prediction	LSTM, attention	Improves LSTM by spatio-temporal attention	Needs exploration of attention mechanism on other models.
DSTA [12]	Stream-flow prediction in large river basins	LSTM, attention	Improves LSTM and Attention-LSTM models	Exploration for small- and medium basins needed.
stResNet-LUBE [13]	Flow interval prediction	LUBE, ResNet	Better performance compared to other deep learning models	Further validation is required on additional datasets

In China, the numerous small- and medium-sized basins present unique challenges for short-term flood forecasting, primarily due to the lack of diverse monitoring tools caused by budgetary and manpower constraints. Fortunately, rainfall and flow data are

often available for these basins. Therefore, many deep learning models primarily rely on rainfall and flow data for flood forecasting [6–10]. Although these models have achieved state-of-the-art results, there is still room for improvement. These models typically use a fixed window of several hours to a day as input and consider the length of the input window as a hyperparameter. Due to the complexity and variability of flood events, models may struggle to adapt to different rainfall conditions using fixed short-term input data. Although theoretically increasing the length of the input window can improve prediction accuracy, experiments with DSTA [12] and stResNet-LUBE [13] have shown otherwise. This is because increasing the input data may also introduce noise, making it difficult for the model to distinguish important features. Therefore, researchers face the challenge of enhancing the model's sensitivity to historical data without altering the fixed input window.

In BERT [14], MAE [15] and other pre-training models, Masked Language Model (MLM) is employed to learn higher-order semantics, enhancing the model's ability to understand contextual information. The main idea of MLM is to randomly mask certain words in the input sentences during training and predict these masked words. By learning to fill in missing information within given contexts, the model enhances its comprehension of contextual details. Inspired by pre-training models, we present the PE-SFPM, designed to enhance the model's ability to perceive historical data through pre-training without altering the input window. The subsequent sections of this paper detail the design, implementation, and evaluation of PE-SFPM, demonstrating its effectiveness and superiority over existing models in flood prediction for small- and medium-sized basins.

3. Methodology

As depicted in Figure 1, our method involves two stages: the pre-training stage and flow forecasting stage. The segment encoder from the pre-training stage will be reused during the flow forecasting stage to enhance flood prediction.

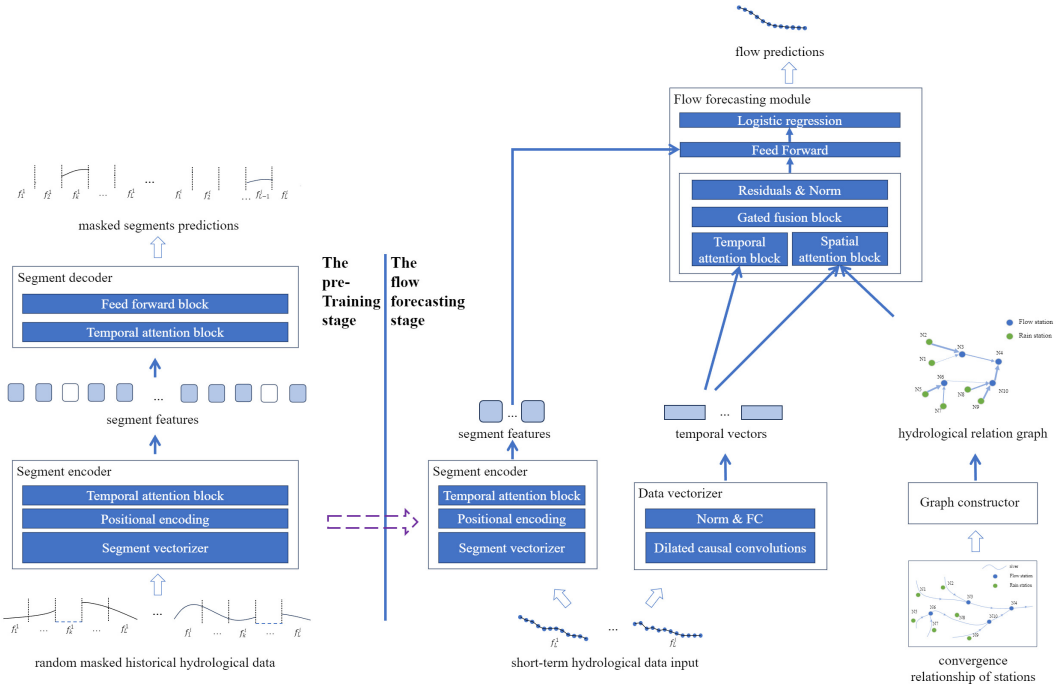

Figure 1. Overall architecture of the proposed method PE-SFPM.

During the pre-training stage, the model utilizes a masking strategy to learn segment features replete with historical features. The model initially splits hydrological data into segments and randomly masks portions of these segments. Subsequently, unmasked segments are fed into a segment encoder, generating corresponding segment features. The

final step involves the use of a segment decoder to parse segment features and predict masked segments.

During the flow forecasting stage, the model employs the segment encoder and a data vectorizer to transform the input data into segment features and temporal vectors. Then, the hydrological relation graph is constructed based on the channel relationships within the basin system. Using the segment features, temporal vectors, and hydrological relation graph as inputs, the flow forecasting module utilizes spatial attention, temporal attention, gated fusion, and feed forward network to complete the flow forecasts.

3.1. The Pre-Training Stage

Drawing inspiration from pre-training models [14–16], our model utilizes an MLM-based strategy for learning historical features during the pre-training stage. Due to the differences between hydrological data and natural language, the design of the pre-training model differs in the following aspects:

1. Each data point in natural language (i.e., words in a sentence) carries rich semantic information, making it suitable as a unit of data for model input. In contrast, individual data points in hydrological data convey much less semantic information, with meaningful semantics emerging at the segment level. Therefore, dividing data into segments encapsulates these semantics, including overarching trends and higher-order meanings.
2. Natural language inherently involves sequential relationships, prompting language models to encode an input sequence based on the entire sequence context. However, hydrological data possess a strict temporal order. To preserve the interpretability of the modeling process, it is critical to consider only past and current information during feature fusion, preventing the leakage of future information. This necessitates masking future features during computation to avoid such leakage.

Based on the above considerations, we propose a pre-training model founded on segment features and a masking attention mechanism. The model mainly comprises a segment encoder and a segment decoder, which are trained through random segment masking and prediction. The procedures involved in the pre-training stage are showcased in the left section of Figure 1.

Prior to constructing the model, hydrological data (rainfall and flow data) are divided into segments and randomly masked. The segmentation size is determined with reference to the input window typically employed in models for short-term flood prediction in small and medium basins. This window usually spans 12 time steps, enabling the direct use of the segments' semantic information during the flow forecasting stage.

The segment encoder processes only partially unmasked data and generating segment features for all the data. With segments as its input units, the encoder initially converts unmasked segments into vectors via a multi-channel convolution. A masking token is used to indicate the presence of segments that are yet to be predicted. Sequential information is then appended to all segments through position encoding. Ultimately, features for all segments are obtained via a masked temporal attention block.

The segment decoder processes all the segment features and predicts those masked segments. The decoder uses a single layer of the temporal attention block to strike a balance between efficiency and performance and employs a feed forward network to determine the expected results of the masked segments.

3.1.1. Segment Vectorizer

In this section, original hydrological data are transformed into vectors through multi-channel standard convolution. More specifically, after dividing the hydrological data into multiple segments, multiple learnable convolutional kernels are applied to unmasked segments to perform convolution operations. The results of all convolutions are then merged to generate feature vectors that represent the segments.

Assume the length of each segment is L, and the data $X = (x_0, x_1, ..., x_s)$ are divided into K segments, satisfying $s = K \times L$. If the embedding dimension for segment features is M, then M convolutional kernels are required, with both the kernel size and stride set to L. Denoting the i-th convolutional kernel as $C^i = (c_0^i, c_1^i, ..., c_L^i)$, the convolution operation executed by kernel C^i on the j-th segment can be expressed as

$$z_j^i = \sum_{l=0}^{L} c_l^i \cdot x_{j*L+l} \tag{1}$$

where z_j^i represents the vector generated after convolution. Hence, the vector for the j-th segment after passing through M convolutional kernels can be expressed as $z_j = (z_j^0, z_j^1, \ldots, z_j^M)$. After vectorization of the segments, the vector Z for data X can be represented as follows:

$$Z = (z_0, z_1, \ldots, z_K) \tag{2}$$

3.1.2. Positional Encoding

Since all the segments are processed simultaneously through convolution, positional encoding is necessary. Positional encoding helps the model to better understand the sequence and positional information within the inputs, enhancing the model's performance and generalization ability. In the case of the Transformer [17], to incorporate positional information into the input features, sine and cosine functions of various frequencies are used to add positional encoding. Assuming that the dimension of the segment vector embedding layer is M, the positional encoding for the i-th segment in the j-th embedding layer dimension can be represented as follows:

$$PE_{i,j} = \begin{cases} \sin(i \cdot \omega_k), & j = 2k..... \\ \cos(i \cdot \omega_k), & j = 2k+1 \end{cases} \tag{3}$$

where $\omega_k = 1/10{,}000^{(2k/M)}$ denotes the frequency of the function, which decreases along the embedding layer dimensions.

Therefore, the positional encoding for the i-th segment can be represented as

$$PE_i = \begin{bmatrix} \sin(i \cdot \omega_1) \\ \cos(i \cdot \omega_1) \\ \sin(i \cdot \omega_2) \\ \cos(i \cdot \omega_2) \\ \vdots \\ \sin\left(i \cdot \omega_{\frac{M}{2}}\right) \\ \cos\left(i \cdot \omega_{\frac{M}{2}}\right) \end{bmatrix} \tag{4}$$

By combining the segment vectors Z with positional encoding PE, we obtain the updated segment vector H, which integrates both feature and positional information, ready for further processing by the neural network:

$$H = Z + PE \tag{5}$$

3.1.3. Temporal Attention Block

Given the temporal dynamic changes implicated in the progression of floods, the significance of feature varies at different time steps, so it is necessary to sieve out key features. It should be noted that features at the current time step can only observe past features without relying on the future. Thus, when attention is calculated, it is imperative to mask the features of future time steps to ensure that the model focuses only on current and historical features.

For the hydrological vector h, it is initially transformed into query, key, and value vectors through multiple multilayer perceptrons (MLPs). The Transformation process can be represented as follows:

$$q = MLP(h), k = MLP(h), v = MLP(h) \tag{6}$$

The message from time step a to time step b can then be defined as follows:

$$m_t^{a \to b} = q^b k^a \tag{7}$$

After calculating the messages between all time steps, the message matrix that contains all messages is summed with the mask matrix to ensure that the current feature does not perceive future features, thus preserving the model's interpretability as follows:

$$M_t = M_t + attnMask \tag{8}$$

where M_t represents the message matrix composed of all features, and $attnMask$ is a matrix with zeros on its main diagonal and below and negative infinity above the main diagonal. Based on the updated message matrix, the attention vector from time step a to time step b can be represented as follows:

$$Att_t^{a \to b} = \frac{Softmax\left(m_t^{a \to b}\right)}{\sqrt{d_b}} v_a, a \in \mathcal{N}_b \tag{9}$$

where $\mathcal{N}_b$ represents the time steps before (and including) b, and d_b represents the embedding dimension of the vector h^b. Based on the attention vectors, all features pointing towards b can be combined to update vector h^b:

$$h^{b_att} = \sum_{x \in \mathcal{N}_b} Att_t^{x \to b} \tag{10}$$

Subsequently, multi-head attention is used to stabilize the learning process and enhance the expression capability of the temporal attention block, obtaining the overall update process for vector h^b:

$$h^{b_multi} = \sigma(h_1^{b_att}, h_2^{b_att}, ...h_E^{b_att}) \tag{11}$$

where the E represents the total number of the attention heads, and the σ function is used to concatenate multiple attention heads and transform them to the same dimension. Finally, the $LayerNorm$ function and residuals are used to stabilize the training process, accelerate network training, and alleviate issues of gradient vanishing and explosion. This process can be described formulaically as follows:

$$h^{b_new} = LayerNorm(h^{b_multi}) + h^b \tag{12}$$

Assuming the length of the hydrological vector h is L, the updated vector h^{new} for the hydrological data can be represented as follows:

$$h^{new} = [h^{1_new}, h^{2_new}, ..., h^{L_new}] \tag{13}$$

3.2. The Flow Forecasting Stage

As illustrated in the right portion of Figure 1, during the flow forecasting stage, the model first generates segment features, temporal vectors, and a hydrological relation graph based on the segment encoder, the data vectorizer, and the graph constructor. Within the flood flow forecasting module, the model leverages both temporal attention and spatial attention to capture temporal features and spatial features. Gated fusion are employed to coordinate the interactions between the temporal and spatial dimensions, ensuring that the prediction results comprehensively reflect immediate information in the short term. Finally,

the model combines segment feature through a feed forward network and utilizes logistic regression to predict.

3.2.1. Data Vectorizer

Dilated causal convolution, a convolutional model for time series data [18], considers only the current and earlier time states during convolution to maintain causal constraints. At the same time, the model effectively reduces the model's depth while expanding the receptive field of the hidden vectors using the dilation factor. In this section, dilated causal convolution is used to extract temporal vectors along the temporal dimension. The vectors are then transformed to specified dimensions using normalization and fully connected layers to acquire an embedded representation of the input data.

Suppose the hydrological data for a station can be represented as $X = (x_0, x_1, \ldots, x_s)$, and the dilation factor is denoted as d. If the convolution kernel is represented as $K = (k_0, k_1, \ldots, k_m)$, then the convolution operation $Conv(x_i)$ for elements x_i can be expressed as follows:

$$Conv(x_i) = \sum_{j=0}^{m} k_j \cdot x_{i-d \cdot j} \tag{14}$$

Based on the normalization function $Norm$ and fully connected layer FC, the overall data X can be further processed and transformed into the vector h. The transforming process can be represented as follows:

$$h = FC(Norm(Conv(X)) \tag{15}$$

3.2.2. Graph Construction

In the task of short-term flood prediction for small- and medium-sized basins, the hydrological data typically comprises flow and rainfall data, which are provided by the stations (rainfall stations and flow stations) within the basin. Flood prediction models need to explore the cause-and-effect mechanisms between rainfall and flow as well as among the flows themselves. These influence relationships can be effectively modeled using a graph. The stations can be represented as nodes, and the relationships between the stations can be represented as edges, and the resulting graph is referred to as the hydrological relationship graph in this paper.

The hydrological relation graph is represented as a directed weighted graph $T = (V, E, W)$, where V denotes the set of nodes, with each node corresponding to a station within the basin, including both rainfall and flow stations; E represents the set of directed edges, indicating the influence between stations; and W signifies the weight function, mapping each edge to a weight value in the interval [0,1], quantifying the association strength between nodes. The choice of [0,1] as the range for weight values is to standardize the expression of relationships between stations, where a weight of 0 indicates no association, and a weight of 1 indicates the strongest influence. To facilitate understanding and manipulation, we further store the hydrological relation graph in matrix form. The matrix's dimensions are consistent with the number of stations within the region, and its elements' values lie within the interval [0,1]. Each element's value reflects the degree of influence one station has on another.

Rainfall in different regions of a basin travels through surface runoff into rivers and converges within these waterways, exerting a direct impact on flow forecasting. Thus, the hydrological relation graph T can be depicted as follows:

$$T = \begin{bmatrix} 1 & \cdots & \frac{1}{r_{1,j}} & \cdots & \frac{1}{r_{1,N}} \\ \frac{1}{r_{2,1}} & \cdots & \frac{1}{r_{2,j}} & \cdots & \frac{1}{r_{2,N}} \\ \frac{1}{r_{i,1}} & \cdots & \frac{1}{r_{i,j}} & \cdots & \frac{1}{r_{i,N}} \\ \frac{1}{r_{N,1}} & \cdots & \frac{1}{r_{N,j}} & \cdots & 1 \end{bmatrix} \tag{16}$$

where $r_{i,j}$ represents the distance between station i and station j, and N represents the number of stations in the basin. Specifically, if i is downstream of j or there is no hydrological upstream–downstream connection between the two stations, $r_{(i,j)}$ should be assigned a numerically high cap or an approximation of infinity. This arrangement ensures that the model appropriately considers the geographical positioning and hydrological interactions of each station.

3.2.3. Spatial Attention Block

In this section, the temporal features are updated based on an attention mechanism, and the influence of these features on flow forecasting is further captured by utilizing the hydrological relation graph. For the feature h_i corresponding to node i, it is first transformed into query, key, and value vectors through multiple multilayer perceptrons (MLPs). The transformation process can be represented as follows:

$$q_i = MLP(h_i), k_i = MLP(h_i), v_i = MLP(h_i) \tag{17}$$

The attention mechanism can be viewed as transmitting messages across a graph. Therefore, the message from node j to node i in the hydrological relation graph is defined as follows:

$$m_s^{j \to i} = q_i k_j + T^{j \to i} \tag{18}$$

where $T^{j \to i}$ denotes the degree of influence between nodes provided by the hydrological relation graph. For the set of nodes $\mathcal{N}_i$, which includes node i itself and nodes that point to node i, let d_i represents the embedding dimension of feature h_i, and the attention vector from node j in $\mathcal{N}_i$ to node i can be represented as follows:

$$Att_s^{j \to i} = \frac{Softmax(m_s^{j \to i})}{\sqrt{d_i}} v_j, j \in \mathcal{N}_i \tag{19}$$

Building upon the attention vectors, the updating process of node i can be represented as

$$h_i^{att} = \sum_{j \in \mathcal{N}_i} Att_s^{j \to i} \tag{20}$$

Finally, multi-head attention is used to stabilize the learning process and enhance the expression capability of the spatial attention block, resulting in the final update process for the feature vector of node i:

$$h_i^{multi} = \sigma(h_i^{att,1}, h_i^{att,2}, ...h_i^{att,Q}) \tag{21}$$

where the Q represents the number of attention heads, and the σ function concatenates multiple attention heads and transforms them into the same dimension as the hidden representation.

3.2.4. Gated Fusion Block

To merge the features from the temporal attention and spatial attention, this section draws on the mechanism found in Gated Recurrent Units (GRUs) [19]. The construction of the fusion gate is based on the outputs of both attention blocks, and the gate is employed to combine the temporal features and spatial features.

Assuming that the output of the temporal attention block is denoted as h_t and the output of the spatial attention block as h_s, the computation process for generating the fusion gate g can be defined as follows:

$$g = tanh(h_t w_t + h_s w_s) \tag{22}$$

where w_t and w_s are learnable parameters, and the $tanh$ function as an additional activation is common for adding non-linearity to the model's learning.

With the fusion gate g established, the process of integrating the fused feature can leverage the Hadamard product (denoted as $\odot$). Therefore, the fused feature h can be calculated as follows:

$$h = g \odot h_t + (1 - g) \odot h_s \tag{23}$$

4. Performance Evaluation

This section initially presents the experimental setup, including the dataset, the evaluation metrics, the baseline models, and the operating environments. It subsequently displays several experimental results, encompassing performance analysis, robustness analysis, and flood ground truth comparison, along with the experimental analysis.

4.1. Experiment Setup

Datasets: To better compare the performance of the proposed method at different levels with baseline models, we select three typical small- and medium-sized basins—ChangHua, HeiHe, and TunXi—as the validation basins for experimental research. The data for these basins, which include both flow and rainfall data, are obtained through STA-LSTM [6].

- ChangHua Datasets: The dataset contains 9354 samples from 8 rainfall stations and 1 flow station, and the period of the dataset is from 7 April 1998 to 20 July 2010.
- HeiHe Datasets: The dataset contains 5423 samples from 10 rainfall stations and 1 flow station, and the period of the dataset is from 1 April 2003 to 10 November 2010.
- TunXi Datasets: The dataset contains 49,532 samples from 12 rainfall stations and 1 flow station, and the period of the dataset is from 27 June 1981 to 18 March 2007.

Metrics: To comprehensively evaluate the performance of the models proposed in this paper, commonly used evaluation metrics for flood prediction were selected, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Baselines: To validate the effectiveness of the proposed model comprehensively, a variety of representative baseline models were used for comparison in the experiments. These included one machine learning-based regression prediction model (SVR [20]), two classic neural network models (LSTM [21] and GCN [22]), and four deep learning-based flood prediction models (Attention-LSTM [8] , ST-GCN [7] , AGCLSTM [9], and DAGAT [10]).

Operating environments: The deep learning models in the experiments are implemented in the PyTorch environment. The training process employs the Root Mean Squared Error (RMSE) as the loss function and the Adam optimizer as the optimizer. The batch size is set to 128, and the initial learning rate is 0.001. The hardware environment used for the experiments is shown in Table 2.

Table 2. Hardware environment configuration.

Component	Specification
CPU	AMD Ryzen 9 3900X @ 3.80 GHz
Memory	64 GB
GPU	NVIDIA GeForce RTX 2070
GPU Memory	8 GB

4.2. Performance Analysis

In flood prediction tasks, predictive accuracy is considered one of the most crucial criteria for evaluating a model. The purpose of performance analysis is to test the model's predictive accuracy and analyze its performance based on various evaluation metrics. In this section, the future 12 h flow is predicted using the past 12 h of rainfall and flow data. The PE-SFPM model is compared with baseline models on various evaluation metrics. Table 3 displays the average performance of all models on the ChangHua, HeiHe, and TunXi datasets.

Table 3. Average performance of all models on the ChangHua, HeiHe, and TunXi datasets.

Dataset	Metric	Model							
		SVR	**GCN**	**LSTM**	**Attention-LSTM**	**ST-GCN**	**AGC-LSTM**	**DAGAT**	**PE-SFPM**
ChangHua	MAE	100.48	89.22	66.16	43.24	39.59	36.16	37.89	**33.68**
	RMSE	189.91	137.60	103.26	75.63	74.09	71.03	68.64	**68.29**
	MAPE	54.60	48.54	38.46	25.6	26.17	24.92	**20.42**	24.3
HeiHe	MAE	86.70	60.23	56.17	38.71	29.46	33.27	24.72	**22.21**
	RMSE	167.68	101.86	100.22	71.82	65.91	68.16	56.87	**56.25**
	MAPE	50.43	33.96	25.76	22.77	20.54	12.71	**9.78**	12.04
TunXi	MAE	100.97	70.25	65.32	58.75	55.45	56.51	**52.80**	53.81
	RMSE	218.17	168.76	140.78	118.9	117.97	118.45	116.28	**107.21**
	MAPE	53.1	33.07	22.33	23.81	24.12	20.36	22.57	**19.82**

In contrast to traditional machine learning approaches such as the Support Vector Regression (SVR) model, which exhibited the lowest performance across the datasets evaluated, deep learning methods have demonstrated substantial superiority in flood forecasting. The subpar results of the SVR model accentuate the gap between conventional regression techniques and the enhanced capabilities offered by deep learning methodologies.

While classic neural network models like LSTM and GCN outperformed SVR, they did not reach the efficacy levels of advanced, specialized flood prediction models. Hydrological data intrinsically possess strong spatial and temporal interdependencies, which carry significant real-world consequences. Models that are finely tuned with domain-specific knowledge typically yield better outcomes. The construction of LSTM and GCN models, although competent, did not fully benefit from the assimilation of flood forecasting expertise, resulting in less than optimal predictive results.

In a detailed comparison, PE-SFPM consistently excels across a spectrum of datasets and evaluation metrics, attaining the highest precision benchmarks amongst the models tested. PE-SFPM incorporates an array of sophisticated optimization strategies, such as pre-training enhancements, convolutions, spatio-temporal attention mechanisms, and gated fusion, to harness and interpret the datasets more effectively; while the individual impact of these varied strategies may differ, their collective implementation invariably contributes to more robust prediction outcomes. It is the integration of these strategies that sets PE-SFPM apart from Attention-LSTM, ST-GCN, AGCLSTM, and DAGAT.

4.3. Robustness Analysis

This part of the evaluation assesses the robustness of the learning effects of various models on different datasets, as well as the variability of their forecast errors. We compare the performance of the PE-SFPM model with that of other baseline models across these datasets. The forthcoming 12 h flow is predicted using the past 12 h of rainfall and flow data. Figures 2–4 illustrate the forecast errors at various time steps for each model.

Analyzing the stability of learning outcomes across a variety of datasets reveals a correlation between the robustness of a model and its accuracy. In general, models specifically designed for flood prediction tend to outperform traditional neural networks. This suggests that the incorporation of domain-specific knowledge into model training not only improves accuracy but also contributes to the model's resilience in different scenarios. In particular, the PE-SFPM showcases commendable performance consistency across diverse datasets, slightly edging out the newly introduced state-of-the-art model, DAGAT, in terms of robustness.

Considering the variability of forecast errors over time, it is noted that the forecast accuracy of all models tends to decrease as the forecast horizon extends. Nevertheless, more robust models demonstrate smaller fluctuations in performance over successive time steps—a fact that is clearly reflected in the gentler gradients of their error trend lines. The PE-SFPM model emerges as particularly resilient in this regard, displaying the most stable performance pattern among the models evaluated. The DAGAT model also exhibits

significant robustness, although marginally less so in comparison to PE-SFPM, while the SVR model is observed to have the lowest level of stability.

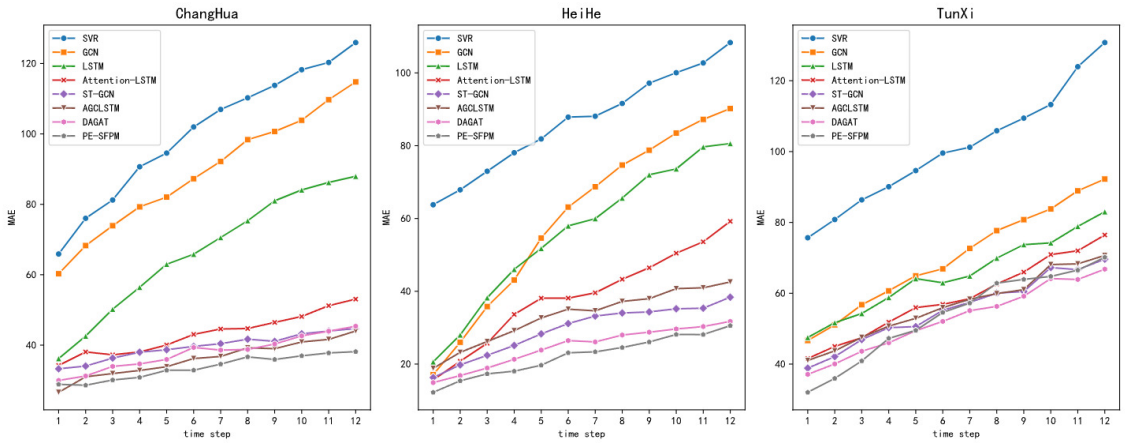

Figure 2. MAE error of the prediction results of each model at multiple time steps.

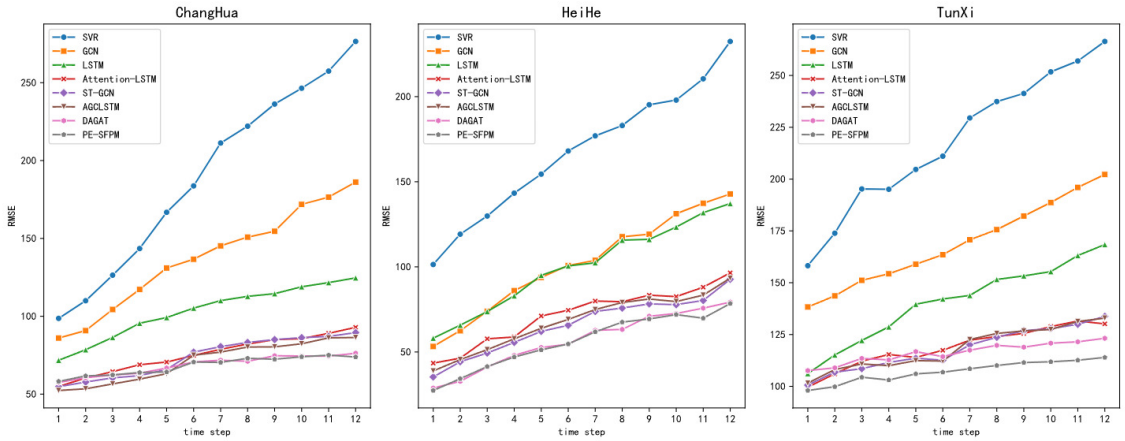

Figure 3. RMSE error of the prediction results of each model at multiple time steps.

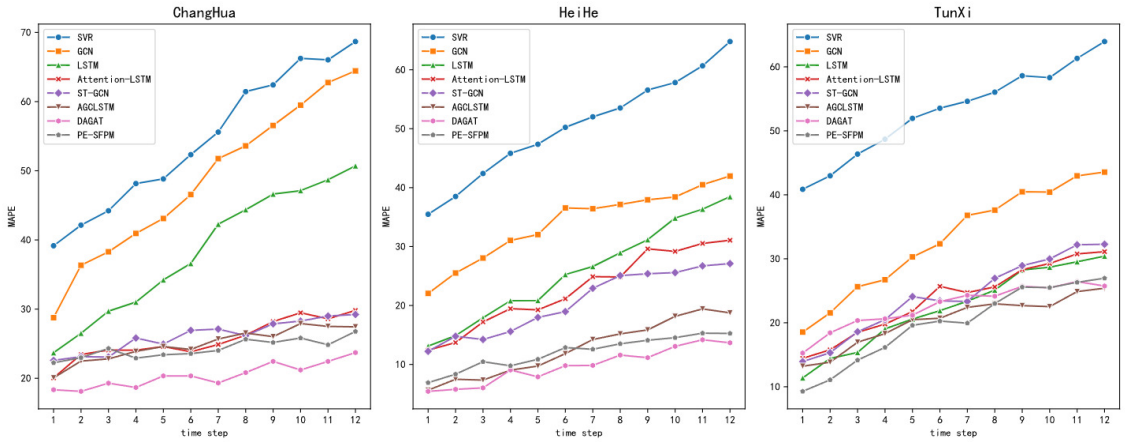

Figure 4. MAPE error of the prediction results of each model at multiple time steps.

4.4. Flood Ground Truth Comparison

In actual flood prediction processes, relying solely on fixed quantitative evaluation metrics may not fully reflect the performance of models. To provide a more intuitive comparison of different models, this section selects actual flow values from the ChangHua, HeiHe, and Tunxi datasets to compare the models' forecasting results for various flood

events. Using 12 h input time window and a forecasting step length of 3 h, the actual forecasts of different flood prediction models for the ChangHua, HeiHe, and Tunxi datasets are shown in Figures 5–7. In these figures, the blue curve represents the actual flow, the red curve represents the PE-SFPM model, the orange curve represents the DAGAT model, and the green curve represents the AGCLSTM model.

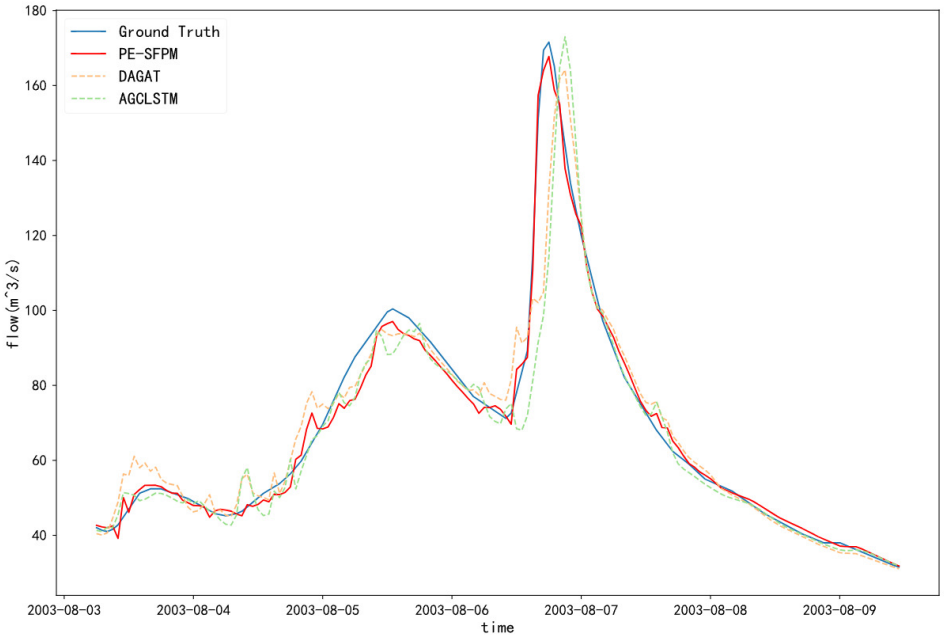

Figure 5. Actual forecasting performance of flood prediction models on the ChangHua dataset.

The AGCLSTM model, which utilizes graph convolutional and spatio-temporal attention mechanisms within its LSTM structure, is adept at capturing spatio-temporal features and their dynamic correlations. Despite this, its performance on the ChangHua dataset indicates a discernible lag in peak flow prediction, suggesting a possible issue with the model's responsiveness to sudden hydrological changes. Furthermore, the observed underestimation of peak flows on the HeiHe and TunXi datasets implies that AGCLSTM might struggle with accurately scaling the magnitude of peak flow.

Similarly, DAGAT applies a sophisticated approach by employing a distribution adaptation mechanism from the Boosting algorithm to calibrate its predictions. Even with this refinement, the model still exhibits a lag on the ChangHua dataset's peak flows. On the HeiHe and TunXi datasets, DAGAT tends to overestimate the peak flows, which could indicate overfitting to certain data features or an inability to generalize well across varied hydrological conditions.

The PE-SFPM model outshines both AGCLSTM and DAGAT on all three datasets. Its predictive accuracy can be attributed to the pre-training strategy, which enhances its capacity to assimilate historical temporal patterns without adjusting the input window. This allows PE-SFPM to efficiently incorporate long-range dependencies, leading to better anticipation of peak flows. Additionally, PE-SFPM's integrated approach, which combines convolutions, attention mechanisms, and gated fusion processes, affords it a nuanced understanding of the data's spatial and temporal facets. This integrated approach helps to accurately gauge the impacts at different times and from various stations, providing a richer and more intricate comprehension of flood dynamics.

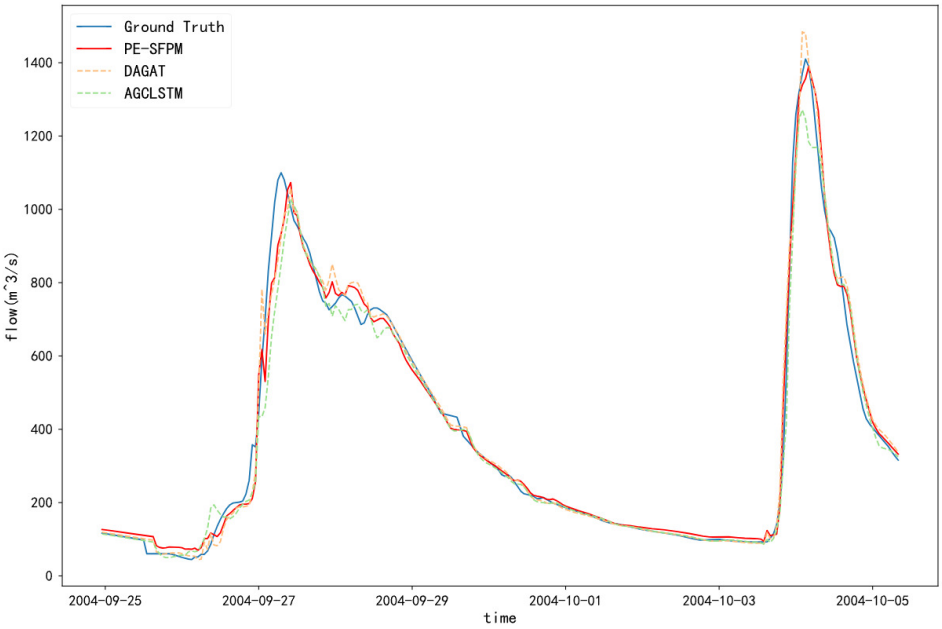

Figure 6. Actual forecasting performance of flood prediction models on the HeiHe Dataset.

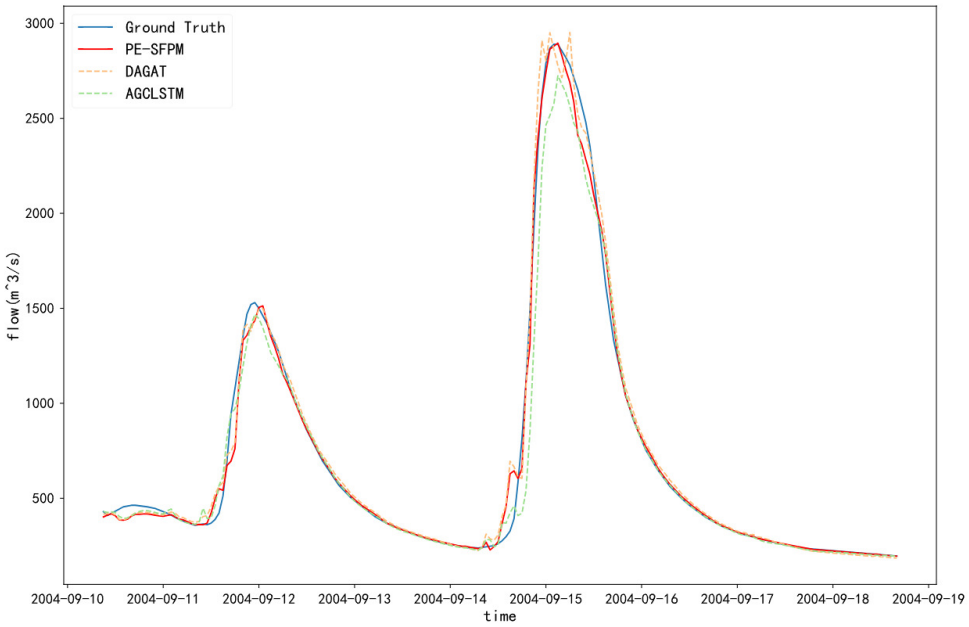

Figure 7. Actual forecasting performance of flood prediction models on the TunXi dataset.

5. Concluding Remarks

In conclusion, we introduce the Pre-training Enhanced Short-term Flood Prediction Model (PE-SFPM). PE-SFPM aims to bolster its sensitivity in recognizing patterns within historical hydrological data, utilizing pre-training methodologies without modifications to the existing input window. Additionally, it incorporates convolution, attention mechanisms, and gated fusion to elucidate and quantify the influences exerted by disparate temporal junctures and distinct measurement stations on flood, succeeding in rendering more precise flood forecasts. The robustness and utility of PE-SFPM have been substantiated via experimentation conducted on three typical basin datasets, and the resulting insights reinforce the viability and superior performance of our proposed model in forecasting flood events.

Training deep learning models is especially challenging for small basins with limited hydrological data. Improving this is a key goal for future research. Upcoming studies will focus on using new methods like transfer learning, few-shot learning, and reinforcement learning. These techniques are expected to help create effective models that can make accurate and reliable flood forecasts, even with small amounts of data.

Author Contributions: Conceptualization, Y.X.; methodology, Y.X.; validation, Y.X.; writing—review and editing, Y.X. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are unavailable due to privacy restrictions.

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

References

1. Kumar, V.; Sharma, K.V.; Caloiero, T.; Mehta, D.J.; Singh, K. Comprehensive overview of flood modeling approaches: A review of recent advances. *Hydrology* **2023**, *10*, 141. [CrossRef]
2. Kumar, V.; Azamathulla, H.M.; Sharma, K.V.; Mehta, D.J.; Maharaj, K.T. The state of the art in deep learning applications, challenges, and future prospects: A comprehensive review of flood prediction and management. *Sustainability* **2023**, *15*, 10543. [CrossRef]
3. Jiang, C.; Kang, Y.; Qu, K.; Long, Y.; Ma, Y.; Yan, S. Towards a high-resolution modelling scheme for local-scale urban flood risk assessment based on digital aerial photogrammetry. *Eng. Appl. Comput. Fluid Mech.* **2023**, *17*, 2240392. [CrossRef]
4. Shen, D.; Bao, W.; Ni, P. A robust real-time flood forecasting method based on error estimation for reservoirs. *AQUA Water Infrastruct. Ecosyst. Soc.* **2022**, *71*, 518–532. [CrossRef]
5. Feng, W.; Shao, Z.; Gong, H.; Xu, L.; Yost, S.A.; Ma, H.; Chai, H. Experimental and numerical investigation of flow distribution pattern at a T-shape roadway crossing under extreme storms. *Eng. Appl. Comput. Fluid Mech.* **2022**, *16*, 2286–2300. [CrossRef]
6. Ding, Y.; Zhu, Y.; Feng, J.; Zhang, P.; Cheng, Z. Interpretable spatio-temporal attention LSTM model for flood prediction. *Neurocomputing* **2020**, *403*, 348–359. [CrossRef]
7. Feng, J.; Wang, Z.; Wu, Y.; Xi, Y. Spatial and temporal aware graph convolutional network for flood prediction. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8. [CrossRef]
8. Yan, L.; Chen, C.; Hang, T.; Hu, Y. A stream prediction model based on attention-LSTM. *Earth Sci. Inform.* **2021**, *14*, 723–733. [CrossRef]
9. Feng, J.; Sha, H.; Ding, Y.; Yan, L.; Yu, Z. Graph convolution based spatial-temporal attention LSTM model for flood prediction. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8. [CrossRef]
10. Feng, J.; Mao, Y. Distribution-Adaptive Graph Attention Networks for Flood Forecasting. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Jakarta, Indonesia, 15–19 November 2023; pp. 340–352. [CrossRef]
11. Wang, Y.; Huang, Y.; Xiao, M.; Zhou, S.; Xiong, B.; Jin, Z. Medium-long-term prediction of water level based on an improved spatio-temporal attention mechanism for long short-term memory networks. *J. Hydrol.* **2023**, *618*, 129163. [CrossRef]
12. Feng, J.; Yan, L.; Hang, T. Stream-flow forecasting based on dynamic spatio-temporal attention. *IEEE Access* **2019**, *7*, 134754–134762. [CrossRef]
13. Yan, L.; Feng, J.; Hang, T.; Zhu, Y. Flow interval prediction based on deep residual network and lower and upper boundary estimation method. *Appl. Soft Comput.* **2021**, *104*, 107228. [CrossRef]
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805. [CrossRef]
15. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009. [CrossRef]
16. Shao, Z.; Zhang, Z.; Wang, F.; Xu, Y. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 1567–1577. [CrossRef]
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
18. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
19. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259. [CrossRef]
20. Wu, J.; Liu, H.; Wei, G.; Song, T.; Zhang, C.; Zhou, H. Flash flood prediction using support vector regression model in a small mountainous catchment. *Water* **2019**, *11*, 1327. [CrossRef]

21. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907. [CrossRef]
22. Le, X.H.; Ho. H.V.; Lee, G.; Jung, S. Application of long short-term memory (LSTM) neural network for flood prediction. *Water* **2019**, *11*, 1387. [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.