

## Article

# A Transfer Learning Approach Based on Radar Rainfall for River Water-Level Prediction

Futo Ueda <sup>1</sup>, Hiroto Tanouchi <sup>2</sup>, Nobuyuki Egusa <sup>2</sup> and Takuya Yoshihiro <sup>2,\*</sup> <sup>1</sup> Graduate School of Systems Engineering, Wakayama University, Wakayama 640-8510, Japan<sup>2</sup> Faculty of Systems Engineering, Wakayama University, Wakayama 640-8510, Japan

\* Correspondence: tac@wakayama-u.ac.jp; Tel.: +81-73-457-8061

**Abstract:** River water-level prediction is crucial for mitigating flood damage caused by torrential rainfall. In this paper, we attempt to predict river water levels using a deep learning model based on radar rainfall data instead of data from upstream hydrological stations. A prediction model incorporating a two-dimensional convolutional neural network (2D-CNN) and long short-term memory (LSTM) is constructed to exploit geographical and temporal features of radar rainfall data, and a transfer learning method using a newly defined flow–distance matrix is presented. The results of our evaluation of the Oyodo River basin in Japan show that the presented transfer learning model using radar rainfall instead of upstream measurements has a good prediction accuracy in the case of torrential rain, with a Nash–Sutcliffe efficiency (NSE) value of 0.86 and a Kling–Gupta efficiency (KGE) of 0.83 for 6-h-ahead forecast for the top-four peak water-level height cases, which is comparable to the conventional model using upstream measurements (NSE = 0.84 and KGE = 0.83). It is also confirmed that the transfer learning model maintains its performance even when the amount of training data for the prediction site is reduced; values of NSE = 0.82 and KGE = 0.82 were achieved when reducing the training torrential-rain-period data from 12 to 3 periods (with 105 periods of data from other rivers for transfer learning). The results demonstrate that radar rainfall data and a few torrential rain measurements at the prediction location potentially enable us to predict river water levels even if hydrological stations have not been installed at the prediction location.

**Keywords:** water-level prediction; flood; transfer learning; CNN-LSTM; radar rainfall

**Citation:** Ueda, F.; Tanouchi, H.; Egusa, N.; Yoshihiro, T. A Transfer Learning Approach Based on Radar Rainfall for River Water-Level Prediction. *Water* **2024**, *16*, 607. <https://doi.org/10.3390/w16040607>

Academic Editors: Jordan Clark and Karl Auerswald

Received: 9 January 2024

Revised: 5 February 2024

Accepted: 7 February 2024

Published: 18 February 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

River water-level forecasting is important in predicting floods and mitigating disaster losses. Numerous studies have been conducted to improve forecasting accuracy [1]. This issue has become even more important in recent years, as climate change has caused more frequent torrential rains and increased flooding worldwide, especially in Asia [2,3]. For instance, Japan has many steep mountains and large river gradients, so torrential rains can easily cause flooding. In fact, flood damage due to torrential rains has been increasing recently, regardless of the size of the river [4]. While large rivers have hydrological stations that can predict water levels several hours in advance and issue evacuation orders, it is difficult to predict water levels on small and medium-size rivers. However, even for those rivers without hydrological stations, water-level prediction is essential for early evacuation [5,6]. The situation is similar in many small and medium-size rivers worldwide, especially in Asia [7].

Traditionally, river water-level prediction has been conducted based on physical models. Many contributions have been made in the literature to improve prediction accuracy [1]. Since the popularization of classical lump models such as the tank model [8] and the storage function (SF) model [9,10], several distributed models have been proposed. Beven et al. [11] proposed TOPMODEL, which incorporates geographic effects to more accurately simulate flow behavior. Sayama proposed the rainfall–runoff–inundation (RRI)

model [12], a two-dimensional (2D) model that simultaneously analyzes rainfall, runoff, and flood inundation. However, building a model for each river requires the fitting of many parameters based on geographical and geological surveys.

Hydrological modeling using artificial neural networks (ANNs) has recently attracted attention due to its high accuracy and availability [13]. Such data-driven prediction models are valuable because they provide accurate predictions only if past hydrological observations' measurement data are available, without geographical and geological surveys. Barzegar et al. [14] achieved accurate lake water-level forecasting by combining wavelet transforms and neural networks incorporating a one-dimensional convolutional neural network (1D-CNN) and long short-term memory (LSTM) structures. Deng et al. [15] investigated the performance achieved by combining a 1D-CNN and LSTM in forecasting daily runoff with data from 24 rainfall stations in China. Yang et al. [16] predicted water quality with the combination of a 1D-CNN and LSTM. However, these models are not capable of handling 2D geographic data and are, therefore, limited in their abilities.

To utilize two-dimensional geographic information, Chen et al. [17] created a 2D rainfall data matrix by Kriging interpolation from 50 rainfall station measurements collected in Xi County, China, and performed a short-term flood prediction with 2D-CNN and LSTM models. Li et al. [18] performed river flow prediction with a CNN-LSTM model and compared its performance to that of the soil and water assessment tool (SWAT) [19]. Xie et al. [20] proposed an ensemble learning model that combines 1D- and 2D-CNN models for daily runoff prediction. Even more advanced deep learning models have been proposed. Alizadeh et al. [21] and Wang et al. [22] introduced an attention mechanism [23] for stream-flow and water-level prediction, respectively. Although the abovementioned studies achieved high-accuracy hydrological prediction, they relied on data from meteorological or hydrological stations upstream of the prediction location. For many small and medium-size rivers, measurements from such stations are unavailable, and the installation of hydrologic stations is impossible due to significant costs. In practice, however, there have been cases of flooding in small and medium-size rivers without hydrological stations [5].

Utilizing radar rainfall data is one of the most promising possibilities for predicting water levels, even in the absence of upstream stations. Because radar rainfall is usually provided as a 2D mesh, it is processed by 2D-CNNs. Several studies have used radar rainfall data in hydrologic prediction models. Liu et al. [24] presented short-term water level forecasts for Fuzhou City using a hydrologic model with rainfall data to show urban flood risk. Baek et al. [25] predicted water level and water quality from radar rainfall images in Korea using a deep learning model with CNN and LSTM structures. Li et al. [26] proposed a rainfall-runoff model that predicts flows of the Elbe River in Germany from 2D radar rainfall images and the measurements of one upstream hydrological station. However, no study has compared the prediction performance achieved using radar rainfall data with that achieved using data from upstream hydrological stations, and it is not yet known whether radar rainfall can replace upstream hydrological stations in terms of prediction accuracy.

Another issue in flood forecasting in small and medium-size rivers is that floods and other disasters are infrequent and often do not provide a sufficient amount of data for deep learning. A promising technique for this problem is transfer learning [27], in which data from different locations are applied to build a prediction model. Using transfer learning, even if there are few inundation data at the forecast site, a river water-level prediction model can be built using data from other rivers with similar characteristics.

To the best of our knowledge, there is only one study in the literature on river water-level prediction using transfer learning, while transfer learning has been widely applied to many areas of study [27,28]. Kimura et al. [29] built a 2D-CNN-based transfer learning model to predict the water level of a river in Japan using water-level and rainfall data obtained from stations in another river basin. Although this work proves the effectiveness of transfer learning in water-level prediction, transferring between two basins in their

method requires the data of two basins obtained from similar station locations. Moreover, they do not utilize the radar rainfall data.

In this paper, a new transfer learning model is presented, which incorporates a CNN and LSTM to predict water levels from radar rainfall images of several river basins across Japan. By newly defining and introducing a flow–distance matrix (see Section 2.4), the model allows for highly accurate transfer learning with a wide range of data from other river basins. The contributions of this paper are summarized as follows.

- Using the proposed CNN-LSTM model (without transfer learning), it is shown that water-level prediction using radar rainfall images is almost as accurate as using measurements from upstream hydrological stations in a torrential rainfall scenario in a relatively steep river in Japan.
- By introducing the flow–distance matrix into transfer learning and using radar rainfall data from other river basins, we demonstrate that water levels can be predicted several hours in advance with high accuracy.
- Through these two contributions, we show fundamental results indicating that water level prediction would be feasible for medium and small rivers, for which historical flood measurements at the prediction site are scarce.

This paper is organized as follows. Section 2 describes the study area, the collected dataset, and the methods implemented with the proposed neural network model for water-level prediction. Section 3 shows the prediction results, which are discussed in Section 4. Finally, we conclude the work in Section 5.

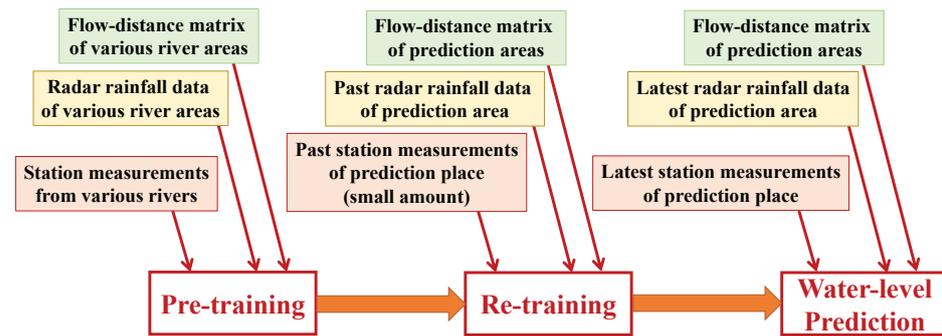
## 2. Materials and Methods

### 2.1. Overview

In this study, transfer learning is applied to predict river water levels using radar rainfall data with a deep learning model incorporating CNN and LSTM structures. As mentioned earlier, this study aims to predict river water levels during heavy rainfall events when there are no measurements from upstream stations. Transfer learning techniques enable us to predict river water levels even when there are few inundation measurements at the prediction point. If flooding due to significantly high water levels is predicted several hours ahead, safe evacuation is possible.

The procedure of river water-level prediction with the required dataset is shown in Figure 1. Two learning steps before prediction are required due to transfer learning. First, the model is pre-trained with historical inundation water-level measurements and the corresponding radar rainfall data of various rivers, where a newly defined flow–distance matrix is also input to make transfer learning work properly. Next, the model is re-trained with the data at the prediction location, i.e., the historical inundation water-level measurements of the station at which water-level prediction will be conducted, as well as the corresponding radar rainfall and the flow–distance matrix. Then, finally, river water-level prediction for several hours ahead is executed with the learned model. Thanks to transfer learning and the radar rainfall dataset, we can predict water levels without upstream stations even if the amount of measurements at the prediction location is small.

The study area is the Oyodo River basin in Japan. The details of the basin and the data are described in Sections 2.2 and 2.3, respectively. As key data to make transfer learning work, the flow–distance matrix is newly defined and introduced as described in Section 2.4. The model of pre-training and re-training is common, incorporating CNN and LSTM structures. The structures of CNN and LSTM are described in Section 2.5, and the proposed prediction procedure, including the model details, is described in Section 2.6. More fundamental, practical, and comprehensive explanations of deep learning, CNNs, LSTM, and transfer learning are available in numerous sources, e.g., Ref. [28].

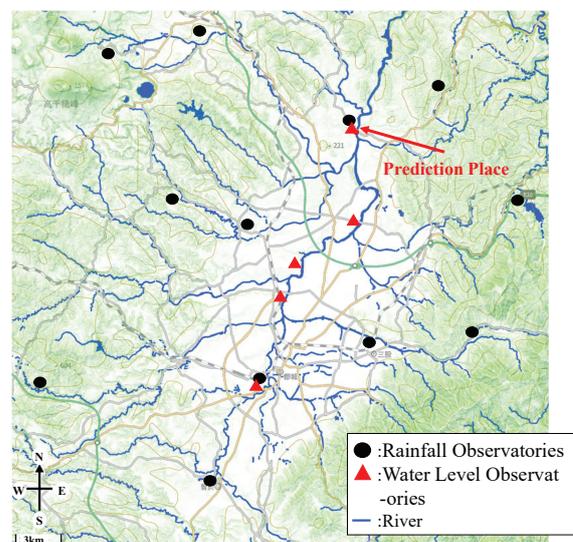


**Figure 1.** Overview of the prediction procedure.

## 2.2. Study Area

In this study, the river water levels at the Hiwatashi hydrological station ( $31.7870^{\circ}$  N,  $131.2392^{\circ}$  E) located in the Oyodo River basin were predicted. Oyodo River is a river in the Miyazaki prefecture, Japan, designated as a class A river. Class A rivers are relatively large rivers managed by national organizations; there are 14,079 class A rivers belonging to 109 class A river systems in Japan. The elevation of the basin ranges from 118 to 1350 m. This area often experiences heavy rainfall due to typhoons. There are 5 water-level observatories and 12 rainfall observatories. The locations of these observatories are shown in Figure 2. Because there are no artificial flow control facilities upstream of the prediction location, this site is well suited for evaluating the accuracy of the predictions of natural water behavior.

The locations of water-level measurements used for pre-training of the transfer learning model are shown in Figure 3. Because the water behavior of the pre-training sites must be similar to that of the prediction sites, we extracted the water-level observatories with similar trends to those of the Hiwatashi Point among the class A rivers in Japan. Specifically, we first identified a 120-h-long period, from 72 h before to 48 h after the peak water-level time, at Hiwatashi Point for the highest peak from 2006 to 2021 (referred to as A). Next, we extracted the periods with higher water-level peaks than the designated water level [30,31] from all the class A rivers in Japan from 2006 to 2021 (referred to as B). Then, we calculated the correlation coefficients between A and B and extracted the periods in B for which correlation coefficients were higher than 0.75, which is a balancing value that ensures high correlation while preserving data volume. As a result, we obtained 105 periods in B as the pre-training data from the 13 class A river systems shown in Figure 3.



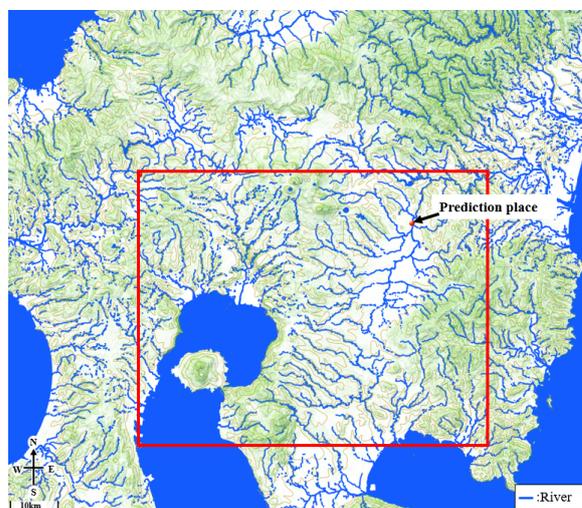
**Figure 2.** Observatories in relation to the prediction site (from the Geospatial Information Authority of Japan [32]).



**Figure 3.** Location of observatories used for pre-training (from the Geospatial Information Authority of Japan [32]).

### 2.3. Data Acquisition

We acquired hourly water-level and rainfall data from the Hydrology and Water Quality Database website of the Ministry of Land, Infrastructure, Transport, and Tourism (MLIT), Japan [33]. We acquired radar rainfall data at 10-min intervals on approximately 1 km mesh (30 s latitude, 45 s longitude) from the Japan Meteorological Agency (JMA) [34]. These data are sold by the Japan Meteorological Business Support Center (JMBSC) [35]. Additionally, we acquired flow direction data, which have an approximate 30 m mesh (latitude 1 s, longitude 1 s), for the entire Japan domain from the website of the Japan Flow Direction Map [36]. The flow-direction data are a matrix, and each cell represents the direction of surface water flow. We call this the flow-direction matrix. Based on the flow-direction matrix, we determined a  $60 \times 60$  km square region to input data into the CNN for each of the pre-training locations and the target location (Hiwatashi). Specifically, the input of the CNN in our model is the radar rainfall data and the flow-distance matrix of the  $60 \times 60$  km region. The method used to create the flow-distance matrix is explained in Section 2.4. The region of the target area, Hiwatashi, is shown in Figure 4. Note that the radar rainfall data have about a 1 km mesh, while the flow-direction matrix is much finer. Therefore, it is necessary to convert the granularity of the flow direction matrix to correspond with the radar rainfall data, as explained in Section 2.4.



**Figure 4.** Area of radar rainfall data (from the Geospatial Information Authority of Japan [32]).

### 2.4. Creating the Flow–Distance Matrix

To predict river water levels using transfer learning, we must overcome the differences among basins, such as flow directions and prediction locations, because the impact of rainfall patterns on the water level at the prediction location is highly dependent on the characteristics of basins. To this end, we defined and created a flow–distance matrix that represents the distance from each to the prediction point of each basin used as the data for pre-training. The flow–distance matrix was created from the flow direction map and input into the CNN model, together with the radar rainfall data. By adding the flow–distance matrix as input, the time delay required for the rainfall to reach the prediction points can be determined, which directly connects the rainfall pattern to the water-level transition at the prediction location and improves the accuracy of predictions.

The flow–distance matrix (data) was created from the acquired flow–direction data with the following operations. The flow direction data are a matrix representing the geographic characteristics of the water flow direction at a granularity of 1-second latitude and 1-second longitude. Each cell in the flow–direction matrix indicates one of the eight directions in which the water flow travels, as shown in Figure 5a. First, the flow directions are converted to the flow distances with the same granularity. We traced the flow direction from each cell to the destination cell, i.e., the cell that includes the water-level prediction location, and computed the distance from the source cell to the destination cell as the number of cells traversed. Here, if the water flow from a cell does not reach the destination, the value ‘null’ is filled in as the distance for that cell. Figure 5b shows an example of this operational step, illustrating the distance from each cell to the destination cell that includes the prediction location in the upper center of the cell.

Next, these flow distances were modified to a distance matrix of coarser granularity to match the granularity of the radar rainfall matrix, which has 30-second latitude and 45-s longitude. This modification was accomplished by taking the average of the  $45 \times 30$  cells of the finer-grained flow–distance matrix shown in Figure 5b. Each average of the  $45 \times 30$  finer-grained cells was put into each cell of the coarser-grained flow–distance matrix, where the geographic boundaries of the matrix conform to the radar rainfall data. If a cell of the coarse matrix includes the prediction location, the value of that cell is set to zero. Figure 5c shows an example of the operation result, where one can see the flow–distance matrix representing the distance from each coarse cell to the prediction location in approximately 1 km mesh.

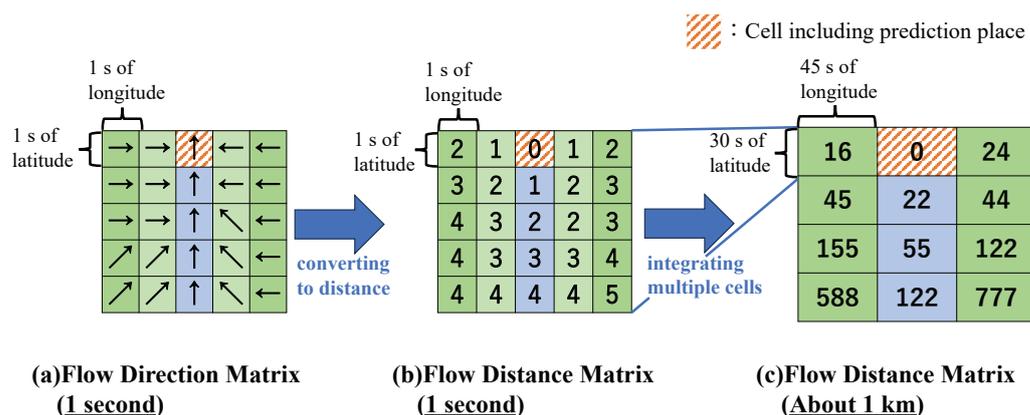


Figure 5. Creating the flow–distance matrix from the surface flow–direction matrix.

### 2.5. Utilized Deep Learning Techniques

#### 2.5.1. Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) use the promising neural network structure to process images [37,38]. A CNN consists of convolution and pooling layers and can learn images without losing spatial features to recognize objects contained in the input images.

The convolution layer applies a convolution operation to create feature maps smaller than the original input image while retaining the feature information to cope with prediction tasks. The convolution operation is a summation-product operation defined by an input image and a filter, as shown in the following Equation (1).

$$u_{ij} = \sum_{p=0}^{W_f-1} \sum_{q=0}^{H_f-1} x_{i+p,j+q} h_{pq}, \quad (1)$$

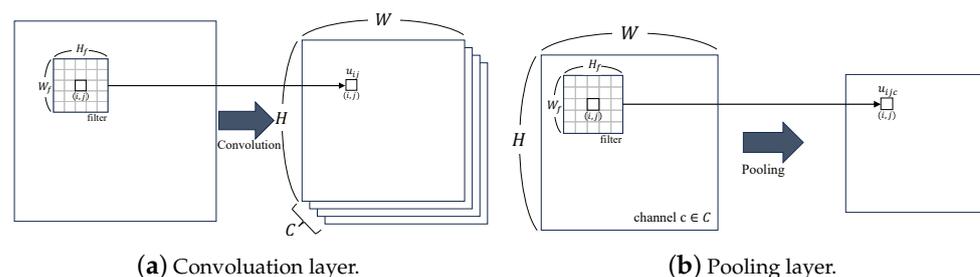
where  $(i, j)$  is the index ( $i = 0, 1, \dots, W - 1, j = 0, 1, \dots, H - 1$ ) of the image,  $x_{ij}$  is the value of a pixel  $(i, j)$  in the image,  $(p, q)$  is the index ( $p = 0, 1, \dots, W_f - 1, q = 0, 1, \dots, H_f - 1$ ) of the filter,  $h_{pq}$  is the value of a pixel  $(p, q)$  in the filter, and  $u_{ij}$  is the value of a pixel  $(i, j)$  in the output feature map. The convolutional layer creates the feature map by the convolution operation shown in Equation (1) for all pixels  $(u_{ij})$  of the output image. The convolution layer usually applies multiple filters and outputs multiple feature maps for each channel ( $c \in C$ , where  $C$  is the set of channels corresponding to the set of applied filters).

The pooling layer is placed after the convolutional layer and performs pooling operations. The pooling layer has two roles [39–42]: One is to summarize the values within a local region of the input, and the other is downsampling, which reduces the spatial resolution of the input. In this study, the pooling layer extracts the maximum pixel value within the filter region with the max pooling operation shown in the following Equation (2).

$$u_{ijc} = \max_{(p,q) \in P_{ij}} z_{pqc}, \quad (2)$$

where  $P_{ij}$  is the filter region for position  $(i, j)$ ,  $(p, q)$  is the index ( $p = 0, 1, \dots, W_f - 1, q = 0, 1, \dots, H_f - 1$ ) within  $P_{ij}$ , and  $z_{pqc}$  represents the pixel values within  $P_{ij}$ .

Figure 6 illustrates the operations in the CNN. Figure 6a shows the convolution operations expressed by Equation (1), where each pixel value  $(u_{ij})$  is calculated according to the filters and the source 2D image is converted to multiple feature maps that shrink according to the filter size. Figure 6b shows the pooling operations. Similar to the convolution layer, the pooling operation calculates each pixel  $(u_{ijc})$  of the output feature maps for each channel ( $c \in C$ ) according to the specified pooling operation (shown by Equation (2) in the case of max pooling).



**Figure 6.** CNN operations.

### 2.5.2. Long Short-Term Memory (LSTM)

In deep learning, a feed-forward network that propagates signals in one direction from the input layer to the output layer without recurrent connections from output to input is called a multilayer perceptron (MLP). In contrast, a recurrent neural network (RNN) is a structure that processes sequential data by considering the temporal information inherent in the sequential data. By feeding the output of the hidden layer at time  $t - 1$  as its own input at time  $t$ , the RNN temporarily stores information acquired from past inputs and reflects it in the final output, allowing it to consider time-series features.

LSTM [43,44] is a well-established model that extends RNNs by replacing the hidden-layer units of RNNs with memory units to allow for long-term memory. The LSTM memory unit consists of three gates for information control (the input gate ( $i_t$ ), forget gate ( $f_t$ ), and output gate ( $o_t$ )) and a memory cell ( $C_t$ ) that holds long-term information. Formally, the structure of LSTM is described in the following equations.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \quad (4)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \quad (5)$$

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (6)$$

$$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \tilde{C}_t, \quad (7)$$

$$h_t = o_t \otimes \tanh(C_t), \quad (8)$$

where  $x_t$  is the input at time step  $t$ ;  $h_t$  is the output of the hidden layer;  $\sigma$  is the sigmoid function;  $\tanh$  is the tanh function;  $\oplus$  is element-wise addition;  $\otimes$  is element-wise multiplication;  $\tilde{C}_t$  is the cell that controls the input gate;  $W_{ij}$  is the weight, where  $i$  represents the input type ( $h$  for the hidden layer and  $i$  for the input layer) and  $j$  represents the gate type; and  $b_j$  is the bias with gate type  $j$ .

The structure of the memory unit in LSTM is illustrated in Figure 7, where Equations (3)–(8) are combined to memorize features. The input of the memory unit for time  $t$  comes from the unit of the previous time step ( $t - 1$ ), and  $C_t$  works to memorize information in the long term, while  $h_t$  memorizes information in the short term. The forget gate ( $f_t$ ) adjusts the level of memory reduction of  $C_t$ , the input gate ( $i_t$ ) adjusts the level at which the information ( $\tilde{C}_t$ ) from the short-term memory is injected into the long-term memory, and the output gate ( $o_t$ ) adjusts the short-term output level. The time-series chain of the memory units selectively maintains the time-dependent feature over time for high-accuracy prediction.

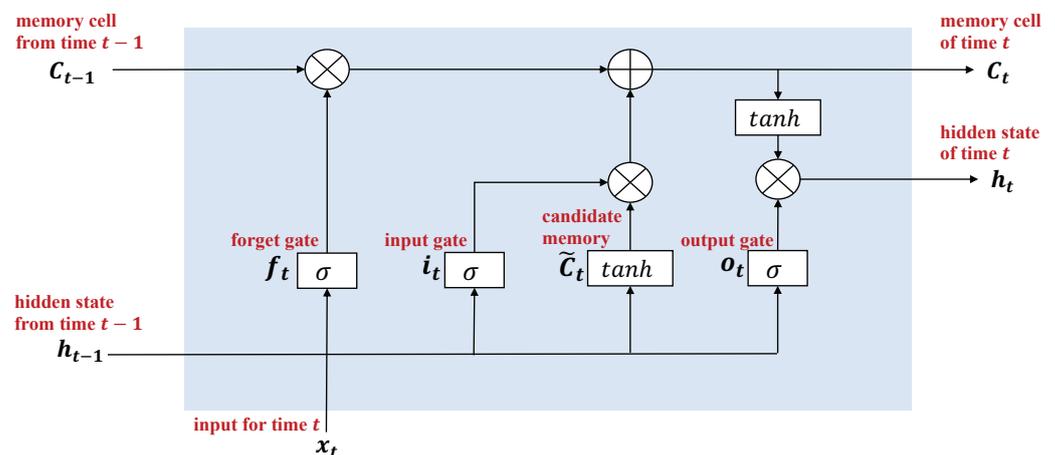


Figure 7. LSTM structure.

### 2.5.3. Transfer Learning

Deep neural networks typically require large amounts of training data. However, when predicting river water levels in anticipation of flooding, there are often few historical flood data available. As a method that can learn with high accuracy even when the amount of data is limited, transition learning is a promising technique [27,28]. Transfer learning is a method that utilizes knowledge from a certain task to learn another similar task. In particular, transfer learning is used for tasks that deal with images, such as image classification, which requires a large amount of training data.

Suppose there is a target task ( $T$ ) and another task ( $T'$ ) similar to it, where the amount of training data ( $D$ ) for  $T$  is limited, but the training data ( $D'$ ) for  $T'$  is abundant. Transfer learning first learns a network for task  $T'$  from  $D'$  (pre-training) and subsequently learns the same network from  $D$  for the target task ( $T$ ) (re-training). Generally, even if the amount of training data ( $D$ ) is limited, if the tasks ( $T$  and  $T'$ ) have a certain degree of similarity, we would achieve high-accuracy prediction for task  $T$  from a rich data set  $D'$ .

This paper presents a new variant of transfer learning for river water-level prediction, while variations of transfer learning have already been presented [27,28]. To apply transfer learning to river water-level prediction, we must overcome the difference in the water-flow characteristics among different basins. The most important factor in water-level prediction in the case of torrential rain is the time that rainwater from each location takes to reach the prediction location. However, precise estimation of water movement requires accurate geographical and geological survey and analysis. Thus, in this paper, the flow–distance matrix, which expresses the distance from each location to the prediction location along the water-flow trajectory, is used to approximate the water travel time. As described in Section 2.4, the flow–distance matrix is created from the flow–direction matrix, which can be created from digital elevation models (DEMs) such as HydroSHEDS [45], which is available to the public. By complementing the difference in characteristics among river basins using the flow–distance matrix, this paper shows that transfer learning is applicable to river water-level prediction.

## 2.6. River Water-Level Prediction Model

### 2.6.1. The Basic Structure Combined with a CNN and LSTM

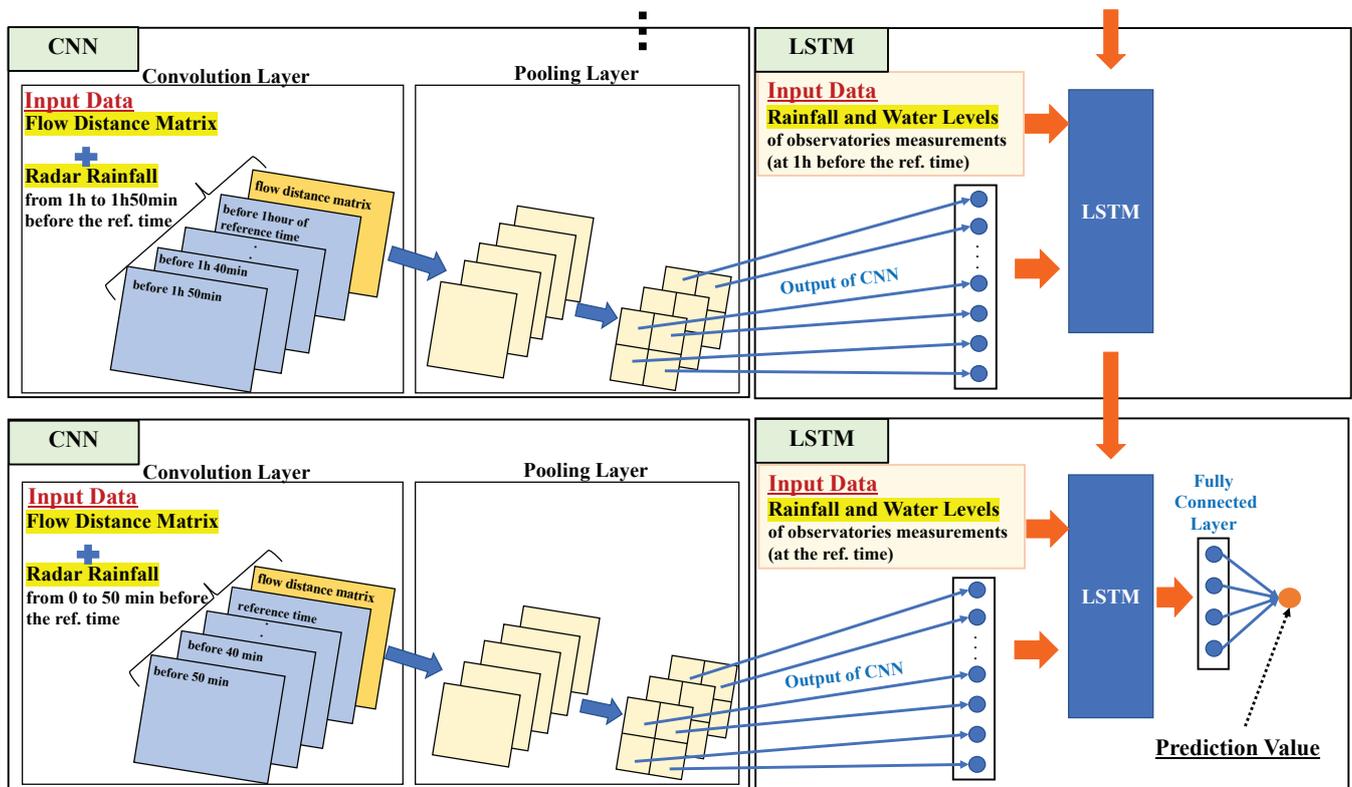
In this study, river-water levels were predicted by a prediction model combined with a CNN and LSTM. An overview of our model is shown in Figure 8. First, the radar rainfall data and the flow–distance matrix are input to convolution and pooling layers to compress the information. Next, the compressed information and the measurements (rainfall and water-level values) from the upstream stations and forecast sites are input into the LSTM layer. The outputs of the LSTM layer are input to the fully connected layer, which outputs the water-level prediction value. Since the measurement interval at the water-level stations is one hour, the time steps in the LSTM layer are at one-hour intervals. Therefore, the radar rainfall data, which are at 10-min intervals, are grouped into hourly time series, and these six channels are input at each time step of LSTM through CNN compression. As a result, a total of seven channels (six channels of radar rainfall data and one channel of the flow–distance matrix) are input to the convolution and pooling layers, and the CNN output for each of the seven channels and the observatories' measurements (rainfall and water level values) are input to LSTM at each time step. Because the amount of data used in this study is not large (as it is inundation data), we built a relatively simple prediction model where all CNN, LSTM, and fully connected layers have a single layer.

In the diagram shown in Figure 8, each row represents one step of LSTM, corresponding to one hour, and the arrows represent the flow of values. The CNN layer consists of a convolution layer and a pooling layer, as shown on the left side of each row. The LSTM layer is seen on the right side, into which values are injected from the CNN layer and the observatories' measurement data. In the final step of LSTM, values are input into the fully connected layer, which outputs the water-level prediction values.

### 2.6.2. Our Transfer Learning Operations

Generally, the amount of inundation data is limited because flooding does not frequently occur—usually, at most, a few times for a river in ten years in Japan. To achieve high water-level prediction accuracy, we applied transfer learning using water-level measurements from various rivers in Japan to complement the limitation in terms of the amount of data at the prediction site. We collected a dataset of rivers with similar characteristics to those of the target site and pre-trained a model with them. With the pre-trained model parameters used as the initial values, we additionally trained the model with a limited

dataset of the target site. This is a typical transfer learning operation that takes advantage of both the pre-training data and the target-site data.



**Figure 8.** Proposed model for river water-level prediction incorporating CNN and LSTM structures.

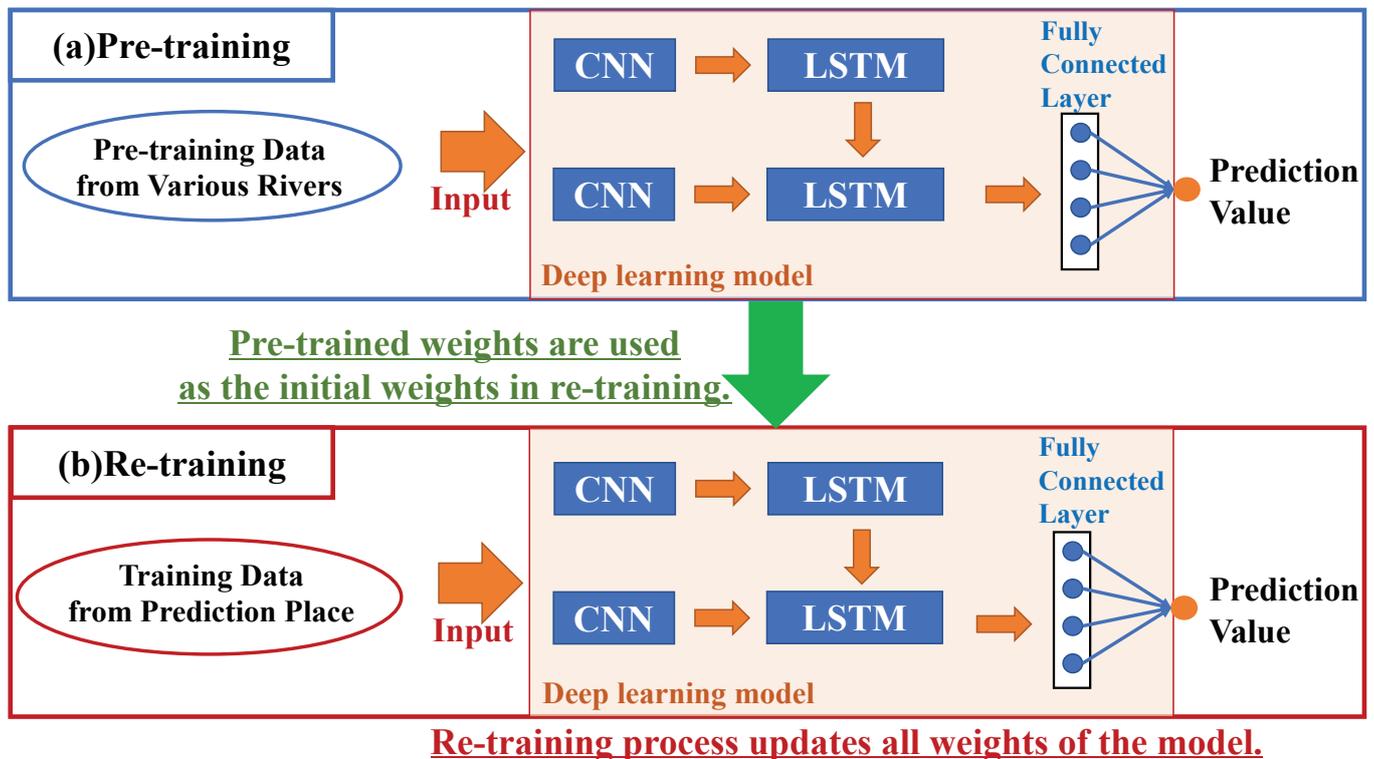
Specifically, the prediction model was pre-trained using the data of the similar-characteristic rivers; the model structure is described in Section 2.6.1. The dataset used here was collected as described in Section 2.2. After that, the model was re-trained using the water-level measurements of the target river; the initial parameter values were the results of the pre-training. All parameters of the neural networks were re-trained because the model is simple, using a single CNN, LSTM, and the fully connected layers. A small learning rate was applied for the re-training because the purpose of re-training weights is the same as ‘fine tuning’, which complementarily improves the model’s performance. By using the flow–distance matrix, the differences in flow directions and characteristics among different rivers can be considered.

Figure 9 illustrates the transfer learning process applied in this study. The upper part (a) shows the pre-training process, where all parameters of the proposed CNN-LSTM-based model are updated with the pre-training data. Next, in (b), the model is re-trained with the data from the prediction site to, again, update all the parameters of the model, using the weights of the pre-trained model as initial weights. This two-stage transfer learning allows for more accurate prediction even when the amount of data available at the prediction site is limited.

### 2.6.3. The Parameter Details

The detailed parameters and structural setup of our model are shown in Table 1. A total of 2000 epochs were executed in the pre-training to ensure that the model was trained with a sufficient amount of data. A small learning rate with the AdamW mechanism [46,47] was used in the re-training to fine tune the weights. To prevent overfitting, the early stopping technique [48] was adopted, where training is stopped if accuracy does not improve in 50 epochs. The general stochastic gradient descent (SGD) method was used

to learn networks, and also the general backpropagation technique was applied to calculate the gradient of the loss function.



**Figure 9.** The sequence of our transfer learning: (a) pre-training with other river data, (b) re-training with the prediction river data.

The input to the CNN layer for each LSTM step is a  $60 \times 60$  km region, as shown in Figure 4. As already mentioned, it consists of seven channels: six radar rainfall channels and one flow–distance channel. Therefore, the input to the CNN is a  $60 \times 60 \times 7$  tensor. The CNN layer consists of a single convolutional layer, a pooling layer, and a dropout layer [49], where a ReLU [50] is used as the activation function. The parameter values in the CNN are shown in Table 2. The LSTM layer consists of a single layer, and the number of units in the hidden layer is set to 100 or 500.

**Table 1.** Structure and parameters of the model.

Item	Parameters
Optimizer	Pre-training: Adam [51] (0.0001) Re-training: AdamW (0.00001)
Epoch	Pre-training: 2000 Re-training: Early stopping (50)
Error function	MSE (Mean Squared Error)
Batch size	Pre-training: 90 Re-training: 50
Language	Python version 3.9.12
Library	PyTorch version 1.12.1

**Table 2.** Parameters of the CNN.

Parameter		Value
Convolutional Layer	Kernel size	$3 \times 3$
	Number of filters	7
	Stride	2
	Activation function	ReLU
Pooling Layer	Classification	Maxpooling
	Window size	$2 \times 2$
Dropout Layer	Pre-training	0.1
	Re-training	0.9

### 3. Results

#### 3.1. Dataset

As mentioned in Section 2.3, the target location for water-level prediction is the Hiwatashi point of the Oyodo River in Japan. The water-level and rainfall data measured at the Hiwatashi station from 2006 to 2021 were obtained, and 13 periods were extracted such that the water-level peaks were higher than the designated water level (5.4 m) [30,31], which is defined for each river observatory by the government. Note that the actual number of such periods is 14, but we excluded 1 because radar rainfall data were missing values. Each of the 13 periods is 120 h long, from 72 h before to 48 h after the peak time in water level. The reason to use only high-peak periods as training data is to avoid the influence of imbalanced data [52]; it is known that deep learning prediction results tend to be biased by characteristics with a large amount of data, and water-level measurement data are mostly dominated in time by normal conditions with low water levels. To predict high-peak water-level transitions, it is desirable to use high-peak period data. The prediction accuracy was evaluated by leave-one-out cross-validation in which data from 12 periods were used to train models to predict the water level for the remaining one period.

It was also previously mentioned that, as the pre-training data for transfer learning, the water-level data of 105 periods of 13 hydrological stations were used, each including 120-h measurements at 1-h intervals. Here, the periods with missing values in their corresponding radar rainfall data were also excluded.

#### 3.2. Evaluation Methods

The objective of our evaluation is to compare the prediction performance with different data sources and prediction models. Specifically, one of our primary concerns is the difference between results obtained with the upstream observatories' measurement data and those obtained with the radar rainfall data instead. Another concern is the performance of transfer learning with the flow–distance matrix incorporating data from other river basins in Japan. To this end, we compared the prediction performance of the following six prediction models, where Model C is intended to be a standard ANN-based prediction model, Model D is the proposed model using radar rainfall data, and Model F is the proposed model with transfer learning.

- A. MLP with the upstream measurement data and the water-level + rainfall data at the prediction location (i.e., Hiwatashi).
- B. LSTM with the upstream measurement data and the water-level + rainfall data at the prediction location.
- C. CNN+LSTM with the radar rainfall data, the upstream measurement data, and the water-level + rainfall data at the prediction location.
- D. CNN+LSTM with the radar rainfall data and the water-level data at the prediction location.
- E. CNN+LSTM with the radar rainfall data, the water-level data at the prediction location, and the flow–distance matrix.
- F. CNN+LSTM, incorporating transfer learning with the radar rainfall data, water-level data at the prediction location, and the flow–distance matrix.

Only prediction Model F uses transfer learning, while the other models do not. Note also that Models A–C use upstream observatories' measurement data, while Models D–F do not.

In the evaluation, the prediction models were trained using values up to 11 h before the reference time for prediction and predicted 1 to 12 h after the reference time. Leave-one-out cross-validation was applied for evaluations, and three criteria were used to evaluate the accuracy of our model: the mean squared error (MSE), Nash–Sutcliffe efficiency (NSE) [53], and Kling–Gupta efficiency (KGE) [54]. MSE, NSE, and KGE are shown in Equations (9)–(11), respectively.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Q_{oi} - Q_{si})^2, \quad (9)$$

$$\text{NSE} = 1 - \frac{\sum_{i=1}^n (Q_{oi} - Q_{si})^2}{\sum_{i=1}^n (Q_{oi} - \mu_o)^2}, \quad (10)$$

$$\text{KGE} = 1 - \sqrt{(r - 1)^2 + (\beta - 1)^2 + (\alpha - 1)^2}, \quad (11)$$

where  $Q_{oi}$  and  $Q_{si}$  are the  $i$ -th measurement and prediction values, respectively;  $r$  is the Pearson correlation coefficient;  $\beta$  is the bias term defined as  $\frac{\mu_s}{\mu_o}$ ;  $\alpha$  is a variability term defined as  $\frac{\sigma_s}{\sigma_o}$ ;  $\mu_o$  and  $\mu_s$  are the averages of the measurement and prediction values, respectively; and  $\sigma_o$  and  $\sigma_s$  are the standard deviations of the measurement and prediction values, respectively.

### 3.3. Parameter Selection

We performed a preliminary evaluation to select the best parameter values for each model (A–F) to fairly compare the ability of the models and the data sources. For Model A, which is based on MLP, ReLU is used as the activation function, and a five-layer network consisting of an input layer, three hidden layers, and an output layer is built with the dropout layer located after each of those five layers. Table 3 shows the attempted combinations of the number of dimensions of the MLP's hidden layers for Model A. Dropout parameters of 0.1 and 0.3 were tested for each of the combinations. We predicted water levels with all combinations of the hidden-layer dimensions and the dropout probabilities and searched for the parameter values that provide the best prediction accuracy.

For Models B–F, which use LSTM, 100 and 500 hidden units were tested. Only for Model B, the prediction accuracy was compared with that of one, two, and four LSTM layers, where dropout was set to 0.3 for the two- and four-layer cases. Other parameters used for Models A–E are shown in Table 4. The parameters for the CNN in Models C–E were the same as those described in Table 2, except that the number of filters in Models C and D was set to six. The parameters of Model F are described in Section 2.6.

The best performance parameters for Models A–F according to our preliminary evaluation are shown in Table 5. For Model B, the best number of LSTM layers was two layers. Those parameter values were used for the evaluation described in the subsequent sections.

**Table 3.** Combination of the number of dimensions of the MLP's hidden layers for Model A.

Combinations of Dimensions
50-30-10
100-50-20
150-100-50
500-300-100
800-400-200
1000-500-100
2000-1000-500

**Table 4.** Parameter settings for Models A–E.

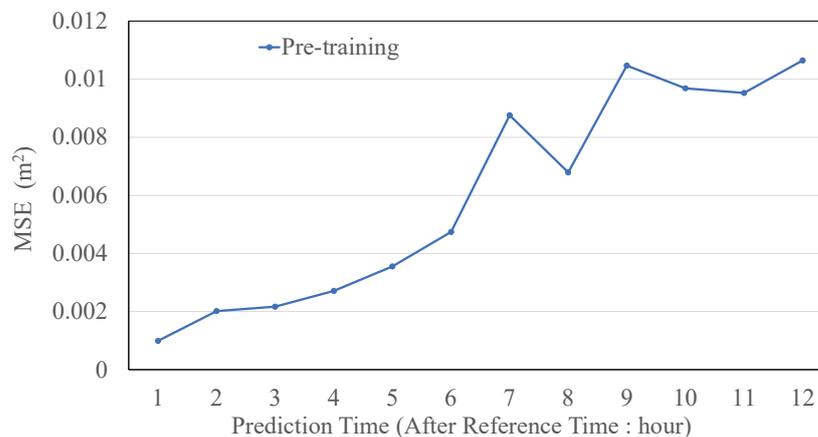
Parameter	Value
Learning Rate	Adam [51] (initial value: 0.0001)
Num. of Epochs	Early Stopping (patience: 50)
Loss Func.	Mean Square Error (MSE)
Batch Size	90
Library	PyTorch

**Table 5.** The attributes and the best performance parameters of the models.

Model	Transfer Learning	Flow–Dist. Matrix	Radar Rainfall	Upstream Data	Dimensions	Drop-Out Probability
A. MLP	no	no	no	yes	50-30-10	0.1
B. LSTM	no	no	no	yes	100	0.3
C. CNN+LSTM	no	no	yes	yes	100	-
D. CNN+LSTM	no	no	yes	no	100	-
E. CNN+LSTM	no	yes	yes	no	100	-
F. CNN+LSTM	yes	yes	yes	no	500	-

### 3.4. Performance in Pre-Training

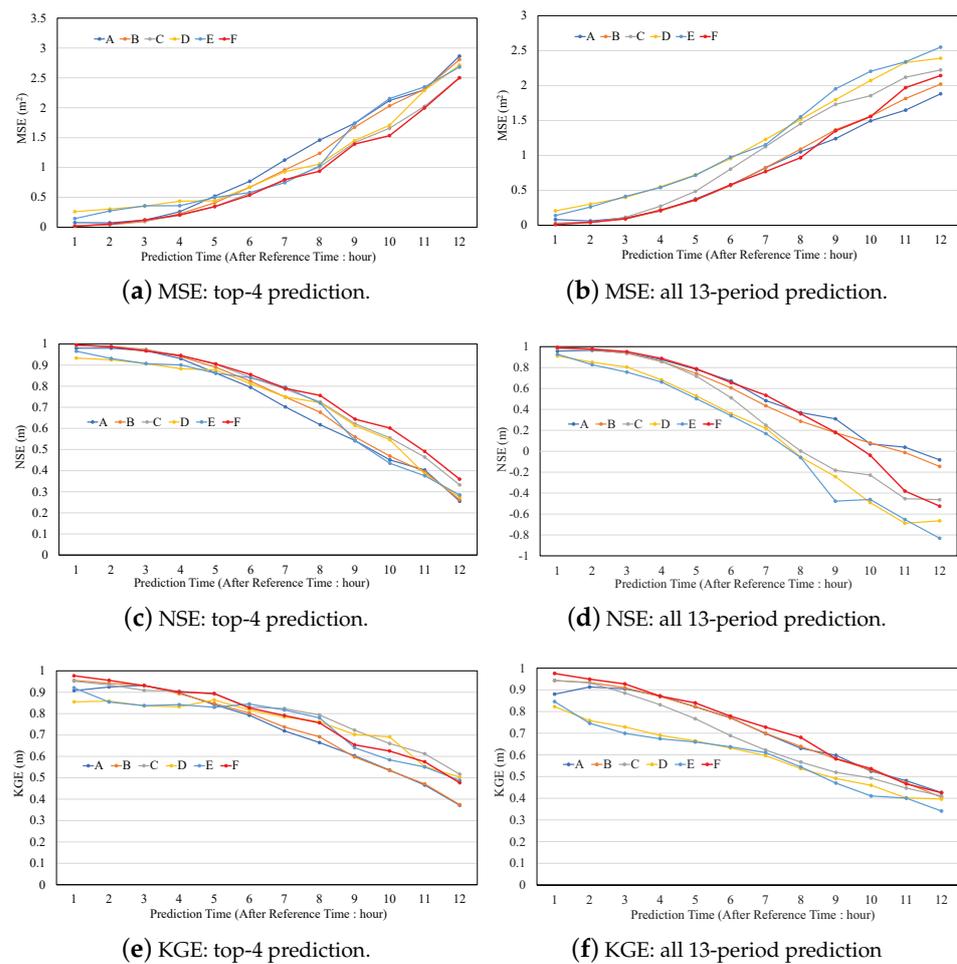
Figure 10 shows the error in MSE in the pre-training of the proposed prediction model (F) for forecasting from 1 to 12 h ahead, where the horizontal axis indicates the prediction time and the vertical axis indicates the MSE, which corresponds to the loss-function values in the training. Figure 10 shows that the MSE of the pre-training is sufficiently small and that the weights of the prediction model were learned with sufficient accuracy using the pre-training data. The 105 periods of water-level measurements used for pre-training are from a wide range of Japanese rivers during inundation, as shown in Figure 3. By incorporating the flow–distance matrix, differences among river basins could be absorbed in the pre-training process.



**Figure 10.** Loss-function values in pre-training.

### 3.5. Performance of the Proposed Method

Figure 11 shows the averaged MSEs, NSEs, and KGEs for each model in forecasting from 1 to 12 h ahead. Because of the large differences in trends, two types of comparisons are presented: the average prediction performance for the top-four peak-water-level periods shown in Figure 11a,c,e and for the average of all 13 periods in Figure 11b,d,f. Note that Models A–C, with which we intend to represent conventional methods, use upstream measurements, and Models D–F do not. In particular, only prediction model F incorporates transfer learning with the flow–distance matrix proposed in this study.



**Figure 11.** The average prediction accuracy.

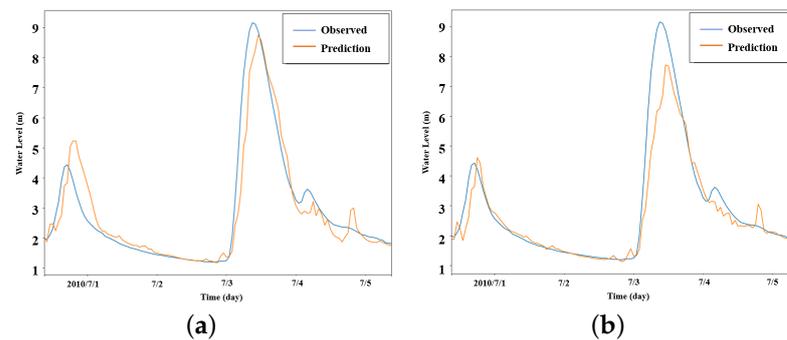
Figure 11a,c,e, for the top-four periods in peak water-level height, show that Model D, which does not use transfer learning and uses radar rainfall data instead of upstream measurements, has a reasonable level of accuracy, although its accuracy drops a little in the 1 to 4 h prediction. Model F, which also does not use upstream measurements but uses transfer learning, improves the accuracy of predictions 1–4 h ahead and can maintain a similar level of accuracy as Model C, which uses upstream measurements. On the other hand, Figure 11b,d,f, for the all 13-period water-level prediction results, show different trends. Models C and D, which use radar rainfall, reduced their performance, and simpler models i.e., Models A and B, based on upstream measurements, performed better. This is because Models C and D, using radar rainfall, tend to overestimate water levels and worked better, particularly for higher-peak cases.

In contrast, Model F, incorporating both the flow–distance matrix and transfer learning, showed good prediction accuracy in both the top-4 and all 13-period cases, although the KGE of the top-4 case is slightly degraded, as shown in Figure 11e. This indicates that the use of the flow–distance matrix for transfer learning is effective in improving the accuracy and the stability of prediction performance.

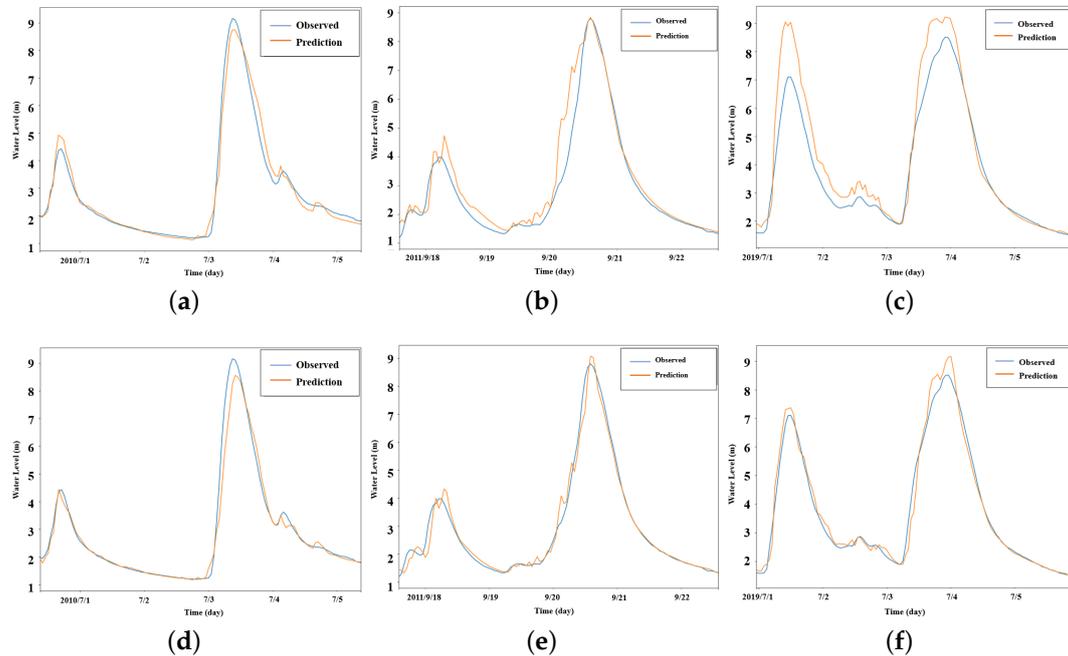
Next, we examine the effect of upstream water-level measurements using hydrographs. Figure 12 shows a hydrograph of the 6 h forecast for the period with the highest peak water level. Figure 12a is the hydrograph of Model C using upstream measurements, and Figure 12b is that of Model F using transfer learning without upstream measurements, where the horizontal axis represents the elapsed time during the period (5 days), and the vertical axis represents the water level in meters. As shown in Figure 11, Model C is slightly more accurate, but this is attributed to the fact that Model C’s predictions are

generally higher than those of Model F. Therefore, although Model F is not as accurate as Model C in the highest water-level part, it is sufficiently accurate in the first peak. The hydrograph shows that Model F can also track water-level changes with reasonably high accuracy.

We further examine the effect of transfer learning on prediction accuracy. Figure 13a–c are the hydrographs of Model D in the 3 h forecast for the top-four peak water-level periods, respectively, and Figure 13d–f are those of Model F. Remember that Model F is more accurate than Model D in the 3-h forecast, as shown in Figure 11. This is because, as shown in Figure 13, Model D predicts higher water levels in some sections, and Model F exhibits a smoother and more accurate predictive trend. This means that the pre-training incorporating the flow–distance matrix effectively improves the accuracy of short-time forecasts by correcting the turbulence in the details of the predictions.



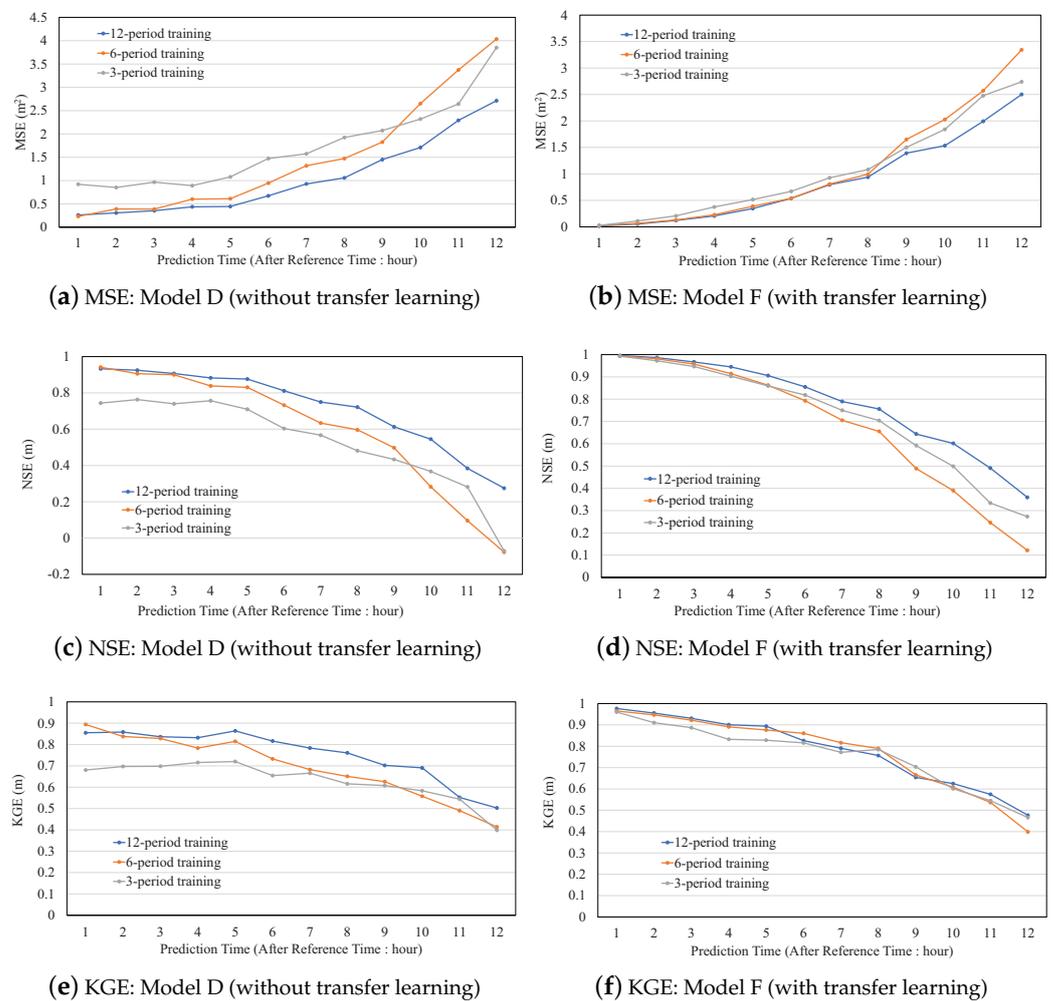
**Figure 12.** Hydrographs for the highest-peak periods. (a) Model C with upstream measurements; (b) Model F incorporating transfer learning without using upstream measurements.



**Figure 13.** Hydrographs of Models D and F for 3 h ahead forecast. (a) The highest-peak periods with Model D (without transfer learning). (b) The second highest-peak periods with Model D (without transfer learning). (c) The third highest-peak periods with Model D (without transfer learning). (d) The highest-peak periods with Model F (with transfer learning). (e) The second highest-peak periods with Model F (with transfer learning). (f) The third highest-peak periods with Model F (with transfer learning).

### 3.6. The Effect of Data Shortage

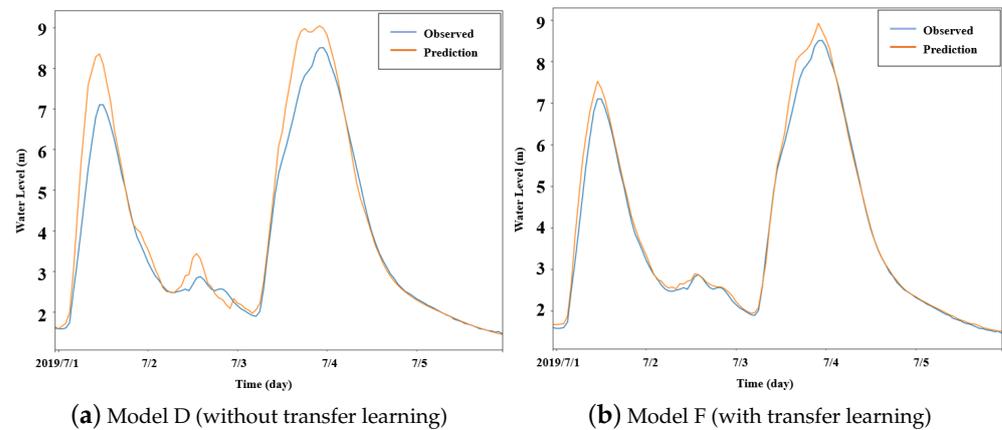
We evaluated the effect of transfer learning by reducing the amount of training data at the prediction point. Specifically, we retrieved top-4 and top-7 peak water-level periods from the 13 periods used in the evaluation and compared the performance through leave-one-out cross-validation (i.e., each uses 3-, 6-, and 12-period data for training). To conduct a fair comparison, however, we compared the prediction performance for the same set of periods; the averages of the top-four peak cases are presented. Figure 14 shows the comparison results between Models D and F, i.e., between models with and without transfer learning. Remember that the prediction accuracy of Models D and F was close in the corresponding top-four cases shown in Figure 11a. In contrast, Model D rapidly decreased in accuracy as the number of training periods decreased, while Model F maintained the level of prediction accuracy for any number of periods. The results show that even when the amount of water-level measurement data at the prediction point is small, introducing data from other rivers through transfer learning enables more accurate water-level prediction.



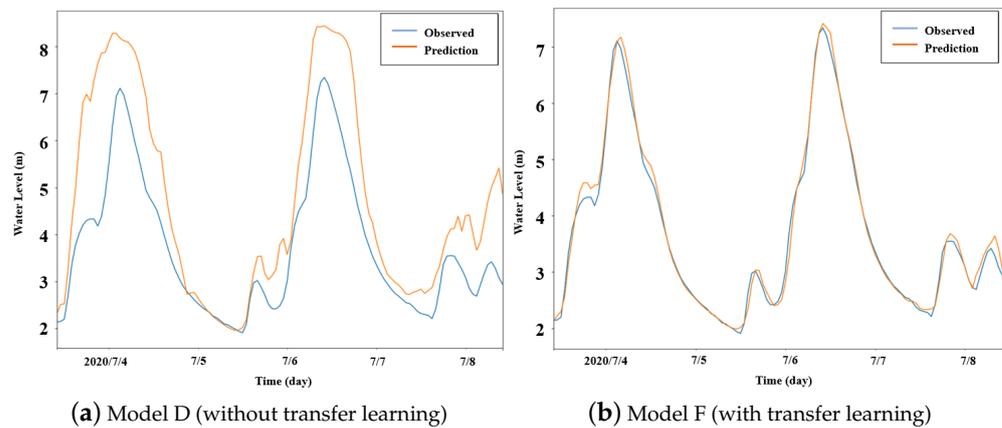
**Figure 14.** Prediction accuracy of Models D and F with various numbers of training periods.

Next, we compared models D and F with their hydrographs. The hydrographs of the 1-h forecast for the periods with the third and fifth highest-peak water levels are shown in Figures 15 and 16, respectively. It can be seen that Model D, which does not use transfer learning, tends to overshoot the predicted water level, while Model F, which uses transfer learning, corrects this tendency and improves the accuracy of the predictions. This also confirms that transfer learning is effective in improving prediction accuracy by

correcting minor errors and fluctuations even when the amount of training data at the prediction point is small.



**Figure 15.** Hydrographs for 1 h ahead forecast (3rd highest period case).



**Figure 16.** Hydrographs for 1 h ahead forecast (5th highest period case).

#### 4. Discussion

The evaluation results show that transfer learning using the flow–distance matrix effectively improves prediction accuracy even when inundation water-level measurement data at the prediction point are scarce. Model F, which incorporates transfer learning without using upstream measurements, achieved a comparable accuracy in water-level forecasting to that of Model C, the conventional CNN-LSTM-based model using upstream measurements. This demonstrates that the proposed transfer learning model that does not require upstream measurements can potentially expand the coverage of rivers to which water level predictions can be applied to mitigate losses due to inundation. There are lots of small and medium-size rivers that do not have upstream observatories, which may cause inundation all over the world. This model, which requires only a few inundation water-level measurements at forecast points, could open the door to developing feasible disaster countermeasures for small- and medium-river inundation.

We see that the flow–distance matrix plays an important role in transfer learning and in predicting river-water levels. Figure 17a shows the values of the loss function in the pre-training of Model F when the flow-distance matrix is excluded from the pre-training input, indicating that excluding the flow–distance matrix in the transfer learning process significantly reduces performance. Figure 17b–d compare the accuracy of water-level predictions with and without the flow–distance matrix, again showing a significant reduction in performance without the flow–distance matrix. Based on those results, we

conclude that the flow–distance matrix is key information for successful transfer learning in river water-level prediction.

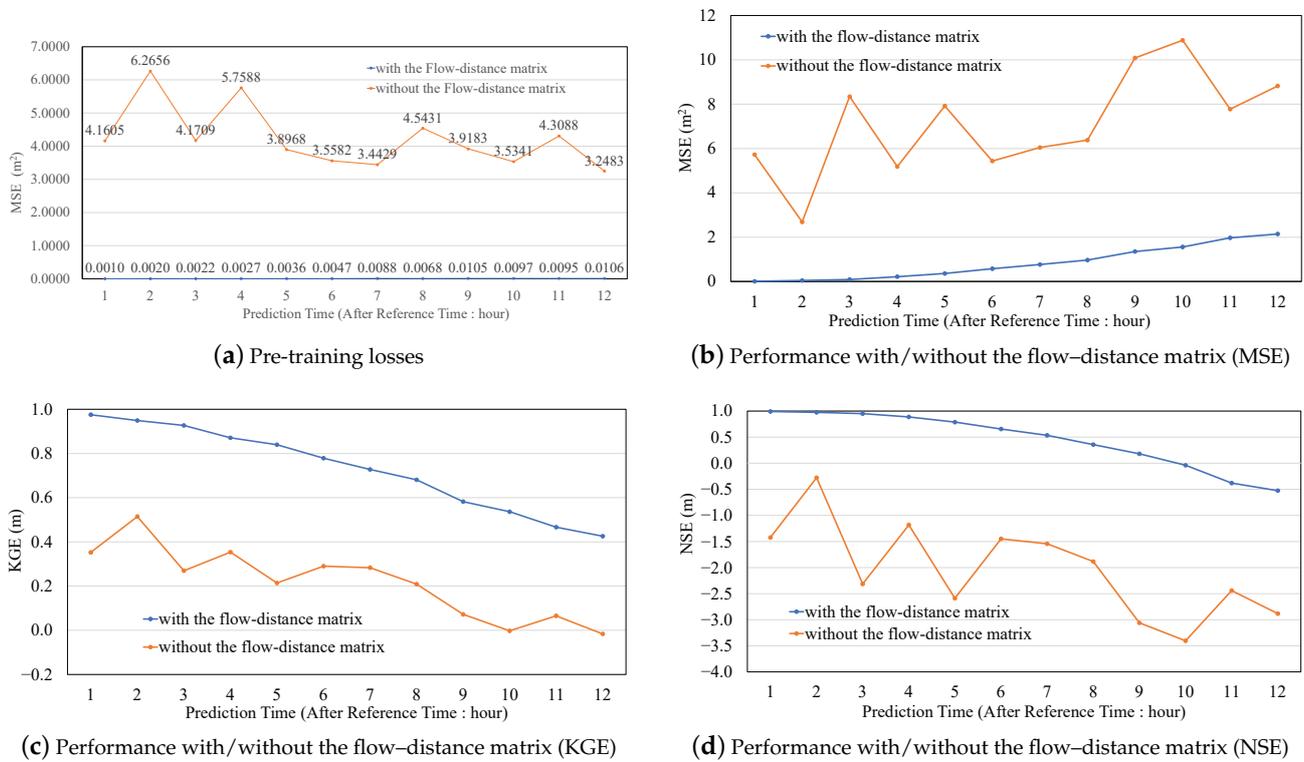


Figure 17. The effect of the flow–distance matrix in transfer learning.

One might think that predictions would be unreliable because of the fluctuation in predicted water levels even in Model F when looking at the hydrographs shown in Figures 15 and 16. This fluctuation in water level is considered to be due to the small amount of input data at the prediction points. However, recall that introducing transfer learning reduced fluctuations and improved accuracy. In other words, increasing the amount of pre-training data improves the accuracy of water-level prediction. Therefore, further increasing the pre-training data from other rivers would solve the problem and possibly improve the accuracy.

In order to mitigate flood damage caused by torrential rains, it is important to accurately predict river levels several hours ahead and issue evacuation advisories at the appropriate time. In many cases, especially for the elderly and disabled, it takes a considerable amount of time to evacuate. Currently, many rivers are at risk of flooding but do not have water-level measurement facilities. Installing even simple water-level sensors may enable the prediction of flooding several hours in advance. While further improvements in models and prediction methods are necessary, there are practical advantages to introducing transfer learning to river water-level forecasting.

It must be kept in mind that this study deals only with the high-water-level cases, although for practical use, it is necessary to provide predictions for the low- and medium-water-level cases as well; people will demand predicted water-level values whenever it rains hard, regardless of whether the water level will be high or low. This will raise the practical challenge of creating a training dataset including variations in water levels while avoiding the imbalance effect of deep learning, i.e., we must create a training dataset by carefully choosing appropriate periods of high, middle, low, and even no elevation in water levels so that the amount of data is not biased by water levels. However, if we only need to distinguish the high-water-level cases from others to issue evacuation advisories, expanding our work by augmenting several middle- or low-water-level cases to cover

a wider range of water levels is enough. This will not be so difficult, because, as shown in Figure 11, our Model F achieved high-accuracy predictions in all cases, with the peak water level ranging from 5.4 to 9.2 m. Training practical models by creating application-specific training datasets is an important future task.

## 5. Conclusions

In this paper, we attempted to predict water levels, anticipating river flooding using radar rainfall data instead of upstream measurement data. Due to climate change, the risk of flooding from torrential rains is increasing, and accurate river water-level forecasting is becoming increasingly important in order to issue evacuation advisories at the proper time. However, the state-of-the-art prediction models require measurements of upstream stations and rich inundation data at the prediction location. To overcome these difficulties, a deep learning model based on a CNN-LSTM structure incorporating transfer learning with the flow–distance matrix was constructed. The CNN-LSTM structure contributes to the utilization of radar rainfall data instead of upstream measurements, and transfer learning reduces the required amount of inundation data at the prediction location.

The evaluation results obtained using inundation data in Japan showed that the presented model can predict water levels several hours ahead with high accuracy using only radar rainfall data and water-level data at the prediction point without using data from upstream stations, with values of NSE = 0.86 and KGE = 0.83, which is comparable to the performance of NSE = 0.84 and KGE = 0.83 of the conventional deep-learning model using upstream station data. Our results also showed that by incorporating the flow–distance matrix, the performance of the transfer learning model with 105 pre-trained data from other rivers was hardly reduced (NSE = 0.82, KGE = 0.82), even when the inundation periods in the training data for the target river were reduced from 12 to 3. These results are significant in that they demonstrate that it is possible to predict river levels even without upstream stations and without abundant data at the target sites, opening the possibility of reducing losses due to flooding in small and medium-size rivers, even without a hydrological station installed.

In future work, improving the prediction accuracy using more inundation data from various rivers in pre-training and evaluating the transfer learning model for rivers of various sizes and characteristics would be interesting for practical use worldwide. One limitation of our method is that it requires proper pre-training data, i.e., water-level data from other rivers with similar characteristics to those of the target river. Developing an efficient method for identifying suitable rivers for pre-training in transfer learning and creating a general pre-training dataset that can be used to predict water levels for any river and valuable future challenges.

**Author Contributions:** Methodology, F.U., H.T. and T.Y.; Software, F.U.; Validation, T.Y.; Resources, H.T.; Data curation, F.U. and H.T.; Writing—original draft, F.U. and T.Y.; Writing—review & editing, N.E. and T.Y.; Supervision, N.E. and T.Y.; Project administration, T.Y.; Funding acquisition, T.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by JST, PRESTO Grant Number JPMJPR1939, Japan.

**Data Availability Statement:** Data sources are all given in Section 2.3.

**Conflicts of Interest:** The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## 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. Asian Disaster Reduction Center. Natural Disaster Data Book 2022 (An Analytical Overview). Available online: <https://reliefweb.int/report/world/natural-disaster-data-book-2022-analytical-overview> (accessed on 26 December 2023).
3. Nihei, Y.; Oota, K.; Kawase, H.; Sayama, T.; Nakakita, T.; Ito, T.; Kashiwada, J. Assessment of climate change impacts on river flooding due to Typhoon Hagibis in 2019 using nonglobal warming experiments. *J. Flood Risk Manag.* **2023**, *16*, e12919. [[CrossRef](#)]

4. World Economic Forum. This Is Why Japan's Floods Have Been so Deadly. Available online: <https://www.weforum.org/agenda/2018/07/japan-hit-by-worst-weather-disaster-in-decades-why-did-so-many-die/> (accessed on 26 December 2023).
5. Council for Social Infrastructure Development. Japan, Report on Rebuilding Flood-Conscious Societies in Small and Medium River Basins, 2017. Available online: <https://www.mlit.go.jp/river/kokusai/pdf/pdf08.pdf> (accessed on 26 December 2023).
6. Kakinuma, D.; Numata, S.; Mochizuki, T.; Oonuma, K.; Ito, H.; Yasukawa M.; Nemoto, T.; Koike, T.; Ikeuchi, K. Development of real-time flood forecasting system for the small and medium rivers. In Proceedings of the Symposium About River Engineering, Online, 27 May 2021. (In Japanese)
7. Trinh, M.X.; Molkenhain, F. Flood hazard mapping for data-scarce and ungauged coastal river basins using advanced hydrodynamic models, high temporal-spatial resolution remote sensing precipitation data, and satellite imageries. *Nat. Hazards* **2021**, *109*, 441–469. [[CrossRef](#)]
8. Sugawara, M. *Rainfall-Runoff Analysis*; Kyoritsu Pub: Tokyo, Japan, 1972; p. 257. (In Japanese)
9. Kimura, T. Storage Function Model. *Civ. Eng. J.* **1961**, *3*, 36–43. (In Japanese)
10. Kawamura, A. Inverse problem in hydrology. In *Introduction to Inverse Problems in Civil Engineering*; Japan Society of Civil Engineers Maruzen: Tokyo, Japan, 2000; pp. 24–30. (In Japanese)
11. Beven, K.J.; Kirkby, M.J. A physically based variable contributing area model of basin hydrology. *Hydrol. Sci. Bull.* **1979**, *24*, 43–69. [[CrossRef](#)]
12. Sayama, T.; Ozawa, G.; Kawakami, T.; Nabesaka, S.; Fukami, K. Rainfall-runoff-inundation analysis of the 2010 Pakistan flood in the Kabul River basin. *Hydrol. Sci. J.* **2012**, *57*, 298–312. [[CrossRef](#)]
13. Karim, F.; Armin, M.A.; Ahmedt-Aristizabal, D.; Tychsen-Smith, L.; Petersson, L. A Review of Hydrodynamic and Machine Learning Approaches for Flood Inundation Modeling. *Water* **2023**, *15*, 566. [[CrossRef](#)]
14. Barzegar, R.; Aalami, M.T.; Adamowski, J. Coupling a Hybrid CNN-LSTM Deep Learning Model with a Boundary Corrected Maximal Overlap Discrete Wavelet Transform for Multiscale Lake Water Level Forecasting. *J. Hydrol.* **2023**, *598*, 126196. [[CrossRef](#)]
15. Deng, H.; Chen, W.; Huang, G. Deep insight into daily runoff forecasting based on a CNN-LSTM model. *Nat. Hazards* **2022**, *113*, 1679–1696. [[CrossRef](#)]
16. Yang, Y.; Xiong, Q.; Wu, C.; Zou, Q.; Yu, Y.; Yi, H.; Gao, M. A study on water quality prediction by a hybrid CNN-LSTM model with attention mechanism. *Environ. Sci. Pollut. Res.* **2021**, *28*, 55129–55139. [[CrossRef](#)]
17. Chen, C.; Jiang, J.; Liao, Z.; Zhou, Y.; Wang, H.; Pei, Q. A short-term flood prediction based on spatial deep learning network: A case study for Xi County, China. *J. Hydrol.* **2022**, *607*, 127535. [[CrossRef](#)]
18. Li, X.; Xu, W.; Ren, M.; Jiang, Y.; Fu, G. Hybrid CNN-LSTM models for river flow prediction. *Water Supply* **2022**, *22*, 4902. [[CrossRef](#)]
19. Arnold, J.G.; Srinivasan, R.; Muttiah, R.S.; Williams, J.R. Large area hydrologic modeling and assessment Part I: Model development. *Am. Water Resour. Assoc.* **1998**, *34*, 73–89. [[CrossRef](#)]
20. Xie, Y.; Sun, W.; Ren, M.; Chen, S.; Huang, Z.; Pan, X. Stacking ensemble learning models for daily runoff prediction using 1D and 2D CNNs. *Expert Syst. Appl.* **2023**, *217*, 119469. [[CrossRef](#)]
21. Alizadeh, B.; Bafti, A.G.; Kamangir, H.; Zhang, Y.; Wright, D.B.; Franz, K.J. A novel attention-based LSTM cell post-processor coupled with bayesian optimization for streamflow prediction. *J. Hydrol.* **2021**, *601*, 126526. [[CrossRef](#)]
22. 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](#)]
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
24. Liu, Y.; Wang, H.; Lei, X.; Wang, H. Real-time forecasting of river water level in urban based on radar rainfall; A case study in Fuzhou City. *J. Hydrol.* **2021**, *603*, 126820. [[CrossRef](#)]
25. Baek, S.; Pyo, J.; Jong, A.C. Prediction of Water Level and Water Quality Using a CNN-LSTM Combined Deep Learning Approach. *Water* **2020**, *12*, 3399. [[CrossRef](#)]
26. Li, P.; Zhang, J.; Krebs, P. Prediction of Flow Based on a CNN-LSTM Combined Deep Learning Approach. *Water* **2022**, *14*, 993. [[CrossRef](#)]
27. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1–40. [[CrossRef](#)]
28. Sarkar, D.; Bali, R.; Ghosh, T. *Hands-On Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras*; Packt Publishing Ltd.: Birmingham, UK, 2018.
29. Kimura, N.; Yoshinaga, I.; Sekijima, K.; Azechi, I.; Baba, D. Convolutional Neural Network Coupled with a Transfer-Learning Approach for Time-Series Flood Predictions. *Water* **2020**, *12*, 96. [[CrossRef](#)]
30. Ministry of Land, Infrastructure, Transport and Tourism (MLIT). List of Water Levels Related to Flood Prevention for Directly Controlled Rivers. Available online: [https://www.mlit.go.jp/river/toukei\\_chousa/kasen\\_db/pdf/2021/12-1-8.pdf](https://www.mlit.go.jp/river/toukei_chousa/kasen_db/pdf/2021/12-1-8.pdf) (accessed on 26 December 2023). (In Japanese)
31. Ministry of Land, Infrastructure, Transport and Tourism. Water and Disaster Management Bureau Implementation Guidelines for Revision of Disaster Prevention Information System for Floods. Available online: [https://www.mlit.go.jp/river/shishin\\_guideline/gijutsu/saigai/tisiki/disaster\\_info-system/](https://www.mlit.go.jp/river/shishin_guideline/gijutsu/saigai/tisiki/disaster_info-system/) (accessed on 26 December 2023). (In Japanese)
32. Ministry of Land, Infrastructure, Transport and Tourism. Geospatial Information Authority of Japan. Available online: <https://www.gsi.go.jp/top.html> (accessed on 26 December 2023).

33. Ministry of Land, Infrastructure, Transport and Tourism (MLIT). The Hydrology and Water-Quality Database. Available online: <http://www1.river.go.jp/> (accessed on 26 December 2023).
34. Japan Meteorological Agency. Nowcast. Available online: [https://www.jma.go.jp/bosai/en\\_nowc/](https://www.jma.go.jp/bosai/en_nowc/) (accessed on 26 December 2023).
35. Japan Meteorological Business Support Center. Available online: <http://www.jmbasc.or.jp/> (accessed on 25 January 2024).
36. J-FlwDir. Japan Flow Direction Map. Available online: <https://hydro.iis.u-tokyo.ac.jp/~yamadai/JapanDir/> (accessed on 26 December 2023).
37. Lawrence, S.; Giles, C.L.; Tsoi, A.C.; Back, A.D. Face Recognition: A Convolutional Neural-Network Approach. *IEEE Trans. Neural Netw.* **1997**, *8*, 98–113. [[CrossRef](#)]
38. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
39. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.A.; LeCun, Y.: What is the best multi-stage architecture for object recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV2009), Kyoto, Japan, 27 September–4 October 2009; pp. 2146–2153.
40. Boureau, Y.L.; Bach, F.; LeCun, Y.; Ponce, J. Learning mid-level features for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2010), Nashville, TN, USA, 20–25 June 2010; pp. 2559–2566.
41. Boureau, Y.L.; Ponce, J.; LeCun, Y. A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th International Conference on Machine Learning (ICML2010), Haifa, Israel, 21–24 June 2010; pp. 111–118.
42. Saxe, A.M.; Koh, P.W.; Chen, Z.; Bhand, M.; Suresh, B.; Hg, A.Y. On random weights and unsupervised feature learning. In Proceedings of the 28th International Conference on Machine Learning (ICML2011), Bellevue, WA, USA, 28 June–2 July 2011; pp. 1089–1096.
43. Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012.
44. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
45. HydroSHEDS. Available online: <https://www.hydrosheds.org/> (accessed on 30 January 2024).
46. Loshchilov, I.; Hutter, F. Fixing weight decay regularization in Adam. In Proceedings of the 6th International Conference on Learning Representations (ICLR2018), Vancouver, BC, Canada, 30 April–3 May 2018.
47. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In Proceedings of the 7th International Conference on Learning Representations (ICLR2019), New Orleans, LA, USA, 6–9 May 2019.
48. Prechelt, L. *Early Stopping—But When?* *Springer Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 53–67.
49. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
50. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on International Conference on Machine Learning (ICMU2010), Haifa, Israel, 21–24 June 2010; pp. 807–814.
51. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 2nd International Conference on Learning Representations (ICLR2014), Banff, AB, Canada, 14–16 April 2014.
52. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. *Learning from Imbalanced Data Sets*; Springer: Cham, Switzerland, 2019; Volume 10.
53. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models part I—A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [[CrossRef](#)]
54. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* **2009**, *377*, 80–91. [[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.