DEVELOPMENT OF A DEEP LEARNING MODEL FOR PREDICTING THE RIVER WATER LEVEL BASED ON THE RAINFALL DISTRIBUTION DATA

YUSUKE NAKATANI
Osaka University, Osaka, Japan, nakatani@civil.eng.osaka-u.ac.jp

MOTOKI OKUMURA
Osaka University, Osaka, Japan, okumura_m@civil.eng.osaka-u.ac.jp

SHUZO NISHIDA
Osaka University, Osaka, Japan, nishida@civil.eng.osaka-u.ac.jp

ABSTRACT

Physical models available for deductively predicting river water levels require a large amount of input data in order to set their boundary conditions and parameters. In this study, we developed a deep learning model comprising a convolutional neural network (CNN) and a recurrent neural network (RNN) with the objective of predicting river water levels based on rainfall distribution data (XRAIN). It was expected that the water level could be predicted using only rainfall data as input data because CNN and RNN can extract spatial and temporal features inherent in the input data, respectively.

The developed model was applied to a case study of the Katsura River basin, Kyoto, and we found that the response of the water level to rainfall could be accurately reproduced. The RMSE was 0.23 m, and the coefficient of determination was 0.91. Nonetheless, with respect to the accuracy of reproduction, a small number of problems, such as unstable outputs in the case of ordinary water levels and an underestimation of peak water levels, could be observed. The accuracy of prediction decreased when the spatial information was not considered as a part of the input data. Furthermore, the accuracy decreased when RNN was not used because, in this case, rainfall history was not considered. The study results demonstrate the necessity of using both CNN and RNN for predicting river water level based on rainfall data.

Keywords: River water level, XRAIN, CNN, RNN, deep learning

1. INTRODUCTION

Physical models that express hydrological runoff processes have been used to predict river water levels, and much research has been undertaken to improve the accuracy of prediction of such models (Mcmillan et al., 2010). The reproducibility of runoff models, however, largely depends on the uncertainties in model parameters for each elementary process, in addition to their initial and boundary conditions. Moreover, if the effects of modeling assumptions and elementary processes that are not considered cannot be ignored, large errors can occur between predicted and observed values. For these reasons, there is a limit to the potential reproduction and prediction of complex hydrological phenomena, including rainfall to runoff, using only physical models. In recent years, research into data assimilation and filtering has been actively conducted (Moradkhani et al., 2005).

On the other hand, the development of machine learning and associated statistical methods in recent years, together with deep learning models using multilayer neural networks (NN), has led to their applications in the hydro engineering fields (for example, Hitokoto et al., 2017; Nakatani et al., 2017). In particular, convolutional neural networks (CNN) used for image recognition and automatic driving of cars (Badrinarayanan et al., 2017), and recurrent neural networks (RNN) used for natural language processing such as machine translation are relatively new methods (Mikolov et al., 2010). Since CNN and RNN can extract, respectively, spatial and temporal features that are implicit in the input data, it may be possible to model hydraulic and hydrological phenomena with strong nonlinearity. By extracting features related to the spatial distribution of rainfall using CNN and extracting features related to rain history and delay time to runoff using RNN, it may be possible to construct a nonlinear model expressing the rainfall–runoff response characteristics of the target basin. Such a model could be used for disaster prevention and as a water level prediction tool in the future.

Many previous studies have used NN to predict river water levels based on a combination of upstream water levels and meteorological data. However, relatively few studies have attempted to predict river water levels using only CNN and RNN from rainfall distributions. It has been reported that NN can predict downstream water levels with high accuracy using the water level upstream of the prediction target point as the input data. In this study, we focus on evaluating the ability of CNN and RNN to express rainfall–runoff processes, and therefore, we use only rainfall data as the input. In Japan, the spatial–temporal distribution data of rainfall is publicly provided by eXtended RAdar Information Network (XRAIN). Since radar rain gauges are widely used around the world, techniques for predicting river water levels based on rainfall distribution alone will certainly become more important in the future.

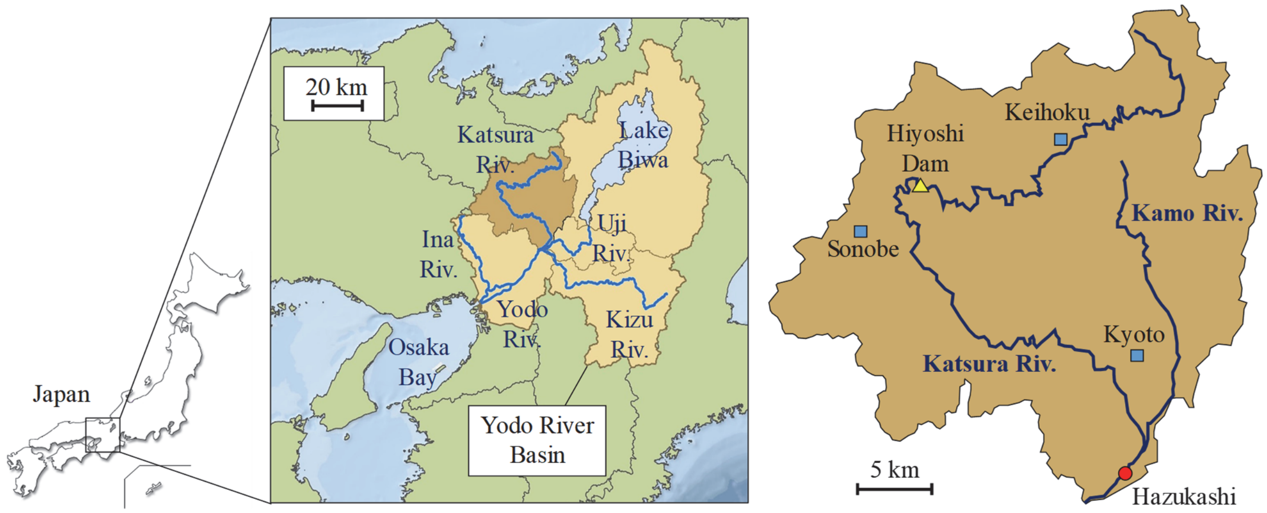
Herein, we develop a deep learning model that predicts river water levels from only radar rainfall data through the application of CNN and RNN. Furthermore, the effects of introducing CNN and RNN in the developed model are examined.

2. METHODS

2.1 Target area

The target area is the Katsura River basin in Kyoto, Japan (Figure 1). The Katsura River is one of the upper tributaries of the Yodo River and has a basin area of approximately 1,100 km^2 and a main channel length of about 114 km. Most of the upper and middle watersheds are surrounded by forest. The middle area is controlled by the Hiyoshi dam, which regulated the flow rate during periods of flooding. In the lower part of the basin, near the urban area of Kyoto, the Katsura River meets with the Kamo River just before the Hazukashi site. Further downstream, the Katsura River joins with the Uji and Kizu Rivers to become the Yodo River and finally flows out to Osaka Bay.

Three meteorological observation sites (Keihoku, Sonobe, and Kyoto) of the Japan Meteorological Agency are located in the Katsura River basin. Figure 2 shows the monthly precipitation at each site averaged from 2010 to 2019. The annual values of precipitation in Keihoku, Sonobe, and Kyoto are 1,848, 1,647, and 1,663 mm, respectively, and precipitation tends to increase toward the north. At each of the sites, there is a clear seasonal variation in precipitation, which is higher in summer and lower in winter.



(a) Yodo River Basin

(b) Katsura River Basin

Figure 1. Study area. The blue squares, yellow triangle, and red circle indicate the meteorological observation sites, the dam site, and the water level observation site, respectively.

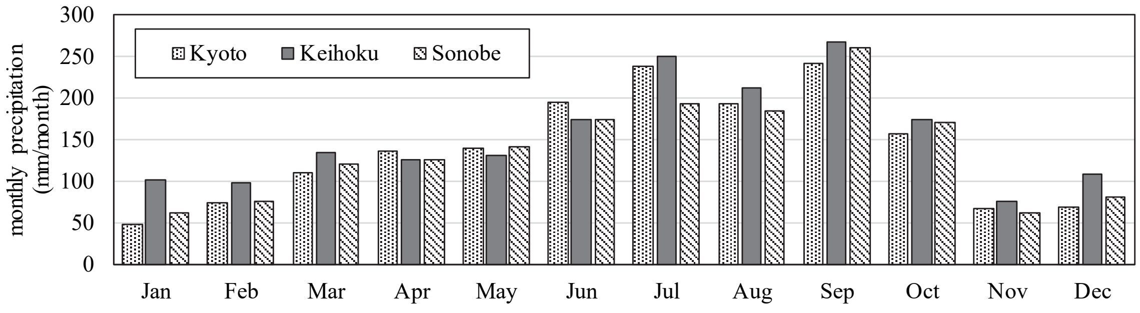


Figure 2. Monthly precipitation (average values for 2010–2019).

2.2 Models

In the following, we explain the predictive model developed in this study and give a brief explanation of CNN and RNN, which are the neural network models constituting the developed model. Please check many other documents for basic explanations related to machine learning, etc.

2.2.1 CNN and RNN

CNN is a neural network model that can mimic the visual cortex system of animals and is often used for image recognition (Krizhevsky et al., 2012). In a simple NN composed of fully connected layers, 2D spatial information input data are lost for the reason that the input data are treated as a 1D array. On the other hand, CNN can extract data features without losing spatial information from the input data by performing local filter operations on multi–dimensional arrays.

CNN is a feedforward-type neural network model comprising a structure in which two convolutional layers and a pooling layer are alternately arranged. In the convolutional layer, a small 2D array, termed the "filter," is moved on the 2D input data, and the correlation between the input data and the filter is calculated at each position. Through this convolution operation, local spatial features within the input data are quantified. Although convolution operations using filters are performed in general image processing (e.g., Gaussian filter and Laplacian filter), CNN can automatically construct optimal filters by teaching these filters to effectively extract the features of the input data. On the other hand, in the pooling layer, processing to reduce the resolution of 2D input data is conducted to increase robustness against noise and spatial differences in the input data. This operation reduces multiple matrix components contained in a small area in 2D input data to one numerical value. Although several pooling methods are available, the max pooling method used in this study outputs a maximum value from multiple matrix components.

Through the filter operation in the convolutional layer, the size of the array of the output data becomes smaller than the input data size as the filter moves in the imaged area. To avoid shrinking of the 2D data, a padding operation to add dummy data to the outer edge of the input data is frequently performed. Additionally, the movement interval of the filter with respect to the 2D input data in the filter operation is often performed, and this process is generally called stride.

RNN is a neural network model that can treat sequential data and is thus often used for natural language processing and sound recognition (Graves and Jaitly, 2014). In RNN featuring a closed loop in the network, the past output of the middle layer can be temporarily stored and reflected in the output at the current time. In this study, we used Long Short-Term Memory (LSTM), a type of RNN able to reflect past input data far from the current time, to output data (Hochreiter and Schmidhuber, 1997). LSTM has a memory unit with a storage capacity in the middle layer and can be utilized for longer time series data than a general RNN.

2.2.2 Predictive model developed in this study

By combining CNN and RNN, we developed a model that can predict river water levels using time series data of rainfall distribution in the basin. In our model, both the spatial distribution of rainfall in the basin and the time delay in the runoff process are considered. Figure 3 shows the structure of the developed model. The input data (2D cumulative rainfall distribution) are shrunk by two CNNs and then pass through a feedforward three–layer NN. Following this process, the data are sent to the RNN and finally output *via* the three–layer NN. In the convolutional layers, padding was performed by adding zero to the outer edge of the data such that the size of the data was not reduced.

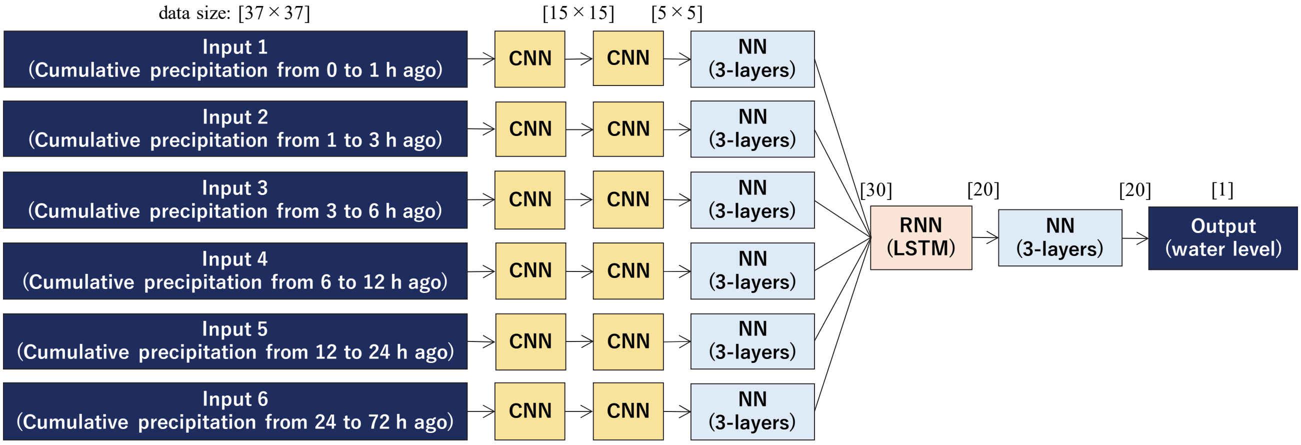

data size: [37 × 37]
[15 × 15]
[5 × 5]
Input 1 (Cumulative precipitation from 0 to 1 h ago)
Input 2 (Cumulative precipitation from 1 to 3 h ago)
Input 3 (Cumulative precipitation from 3 to 6 h ago)
Input 4 (Cumulative precipitation from 6 to 12 h ago)
Input 5 (Cumulative precipitation from 12 to 24 h ago)
Input 6 (Cumulative precipitation from 24 to 72 h ago)
CNN
CNN
NN (3-layers)
[30]
RNN (LSTM)
[20]
NN (3-layers)
[20]
[1]
Output (water level)

Figure 3. Structure of the predictive model.

Table 1. The principal hyperparameters used in the model.

LEARNING RATE	LEARNING ITERATIONS	FILTER SIZE	STRIDE	LOOK BACK TIME
10^{-5}–10^{-3} (Adam)	300 + α	5 km (convolutional layer) 3 km (pooling layer)	2 (convolutional layer) 1 (pooling layer)	168 h

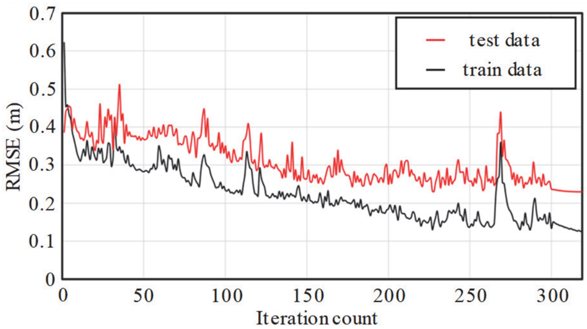


Figure 4. Changes in RMSE after learning (Case 1).

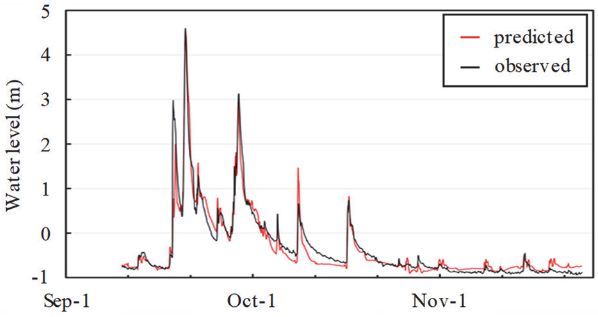


Figure 5. Comparison between observed and predicted time series (Case 1).

Although the model contains many hyperparameters, we tuned several parameters that exert a large effect on model reproducibility to ensure that the observed and predicted water levels match well. The main parameters obtained are shown in Table 1. The error between the predicted and observed values was evaluated according to the root-mean-squared error (RMSE) of the water level.

2.3 Data

Rainfall distribution data from XRAIN were used as the input data, since this dataset has a spatial resolution of 250 m and a temporal resolution of 1 min. Owing to a limitation of the environment of analysis, data converted to 1 km resolution by averaging were used. In addition, the time interval between rainfall data and the river water level data obtained at 1-h intervals were matched: the XRAIN data were piled to hourly data as an accumulated precipitation. The hourly water level data at the Hazukashi site (located approximately 8 km upstream of the lowest point of the Katsura River; see Figure 1) were used as output data. The analysis period selected was between 2015/1/1 and 2017/10/31. It is important to note that both the water level and rainfall data during the analysis period included missing values and error values. Consequently, after excluding these, a total number of 12,800 datasets were used in this analysis. The datasets were divided into train data (11,000 datasets) and test data (1,800 datasets) to evaluate the generalization performance of the developed model.

2.4 Analysis cases

First, as in Case 1, we evaluated the prediction performance of river water level using the developed model. Next, we conducted two calculation cases to evaluate the effect of introducing CNN and RNN as tools for water level prediction. In Case 2, the effect of introducing CNN was evaluated by performing calculations that used a simple averaging filter only once, instead of using the filter trained for the CNN convolution. In Case 3, the effect of introducing RNN was evaluated, and the calculation was performed using a feedforward three–layer NN instead of the LSTM unit.

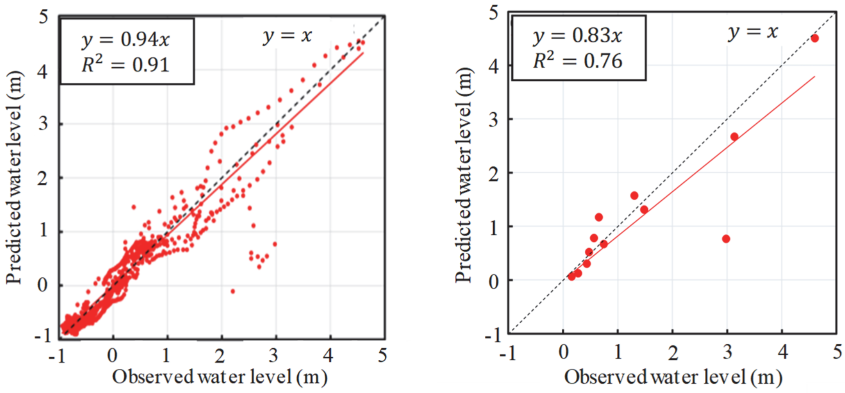
3. RESULTS AND DISCUSSION

3.1 Reproducibility of river water level by the developed model (Case 1)

Figure 4 shows the transition of RMSE due to learning. Although RMSE sometimes increased suddenly, the RMSEs of both the training and test data converged to constant values and finally reached approximately 0.12 and 0.23 m, respectively.

Figure 5 compares the observation and prediction results from the developed model for the time series of river water levels. The predicted values were found to accurately reproduce the fluctuations of the observed values, suggesting that the developed model faithfully represents the rainfall–runoff process.

Figure 6 shows the relationships between the observed and predicted values for (a) all test data and (b) only the data at the peak water level. The red line in the figure is the regression line, and its formula and determination coefficient are also shown. The black dashed line indicates the straight line $y = x$. Predicted values differ significantly from the observed values over the range of 2 to 3 m water levels because of the low



(a) all data (b) only data at peak water level

Figure 6. Relationships between observed and predicted values (Case 1).

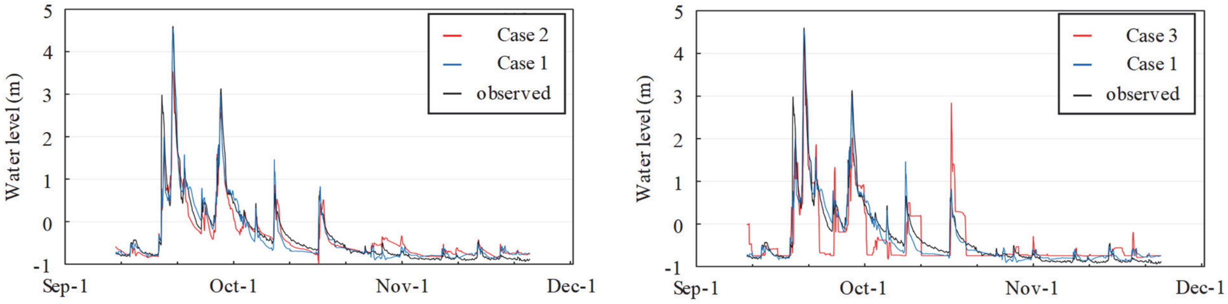


Figure 7. Time series of river water level (Case 2 and Case 3).

reproducibility of the model during the drawdown period after the water level peak. However, since the slope of the regression line is 0.94 and the determination coefficient (R^2) is 0.91, the reproducibility of the model can be deemed high. From the viewpoint of disaster prevention, it is important to accurately predict peak water levels during periods of flooding. As shown in Figure 6(b), the slope of the regression line was 0.83, and its determination coefficient was 0.76, indicating that the model reproduced the peak relatively well. However, since a large error of about 2 m may occur, further improvement in accuracy will be necessary before using this model for flood forecasting.

3.2 Effects of introducing CNN and RNN (Case 2 and Case 3)

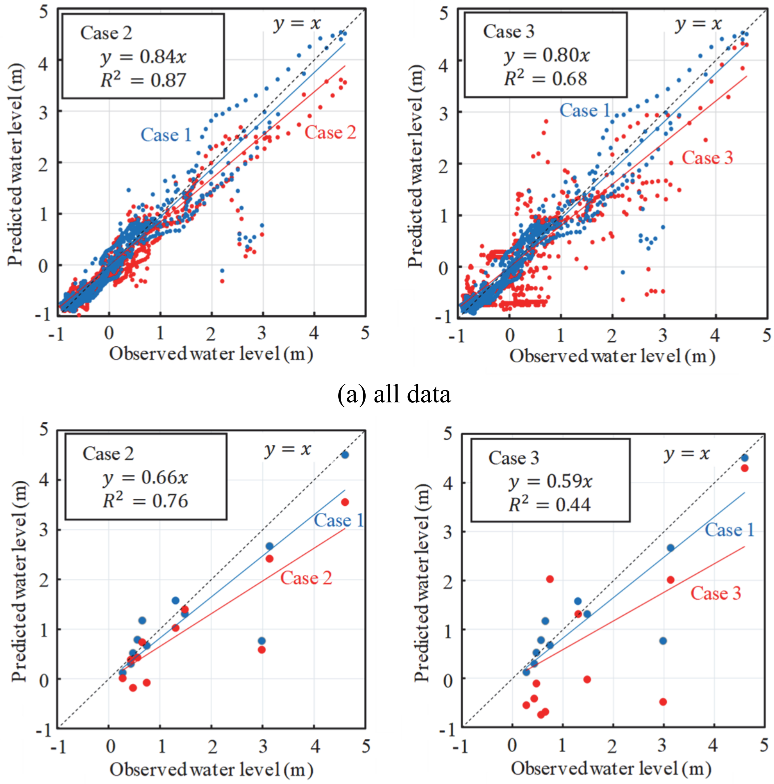
Figure 7 shows the observation data and calculation results for Case 2 or Case 3 of river water level. In addition, Figure 8 shows the relationships between observed and predicted values of river water level.

In Case 2, the predicted values seem to follow the fluctuation of the observed values in a similar manner to that of Case 1. However, the water level was underestimated in Case 2 relative to Case 1. After evaluating the prediction accuracy of the peak water level, the slope of the regression line was found to be 0.66, and the determination coefficient was 0.76, indicating that the prediction accuracy was, indeed, significantly lower than that in Case 1. This suggests that it is important to consider spatial characteristics in rainfall distribution in order to accurately predict the river water level.

In Case 3, on the other hand, although the predicted value responded to the water level rise after rainfall, the subsequent decrease in water level could not be reproduced. Furthermore, water level fluctuations at low water levels were almost linear in the prediction results, and no fine fluctuations were reproduced. Furthermore, in Case 3, the water level tended to be underestimated to a greater extent than that in Case 2. When focusing only on peak water levels, the slope of the regression line was 0.59, and its determination coefficient was 0.44, indicating that the prediction accuracy was extremely low. In Case 3, the RNN layer was replaced by the three-layer NN that does not consider recurrent information. These results indicate that the prediction of water levels is impossible without considering past rainfall history.

4. CONCLUSIONS

In this study, a deep learning model was developed and shown to be able to predict river water level using only radar rainfall data. Applying the model to the Katsura River basin, fluctuations in the river level were successfully reproduced, including the peak levels during periods of flooding, and an accuracy of 0.23 m in RMSE was obtained.



(a) all data

(b) only the data at the peak water level

Figure 8. Relationships between observed and predicted values (Case 2 and Case 3).

The effects of introducing CNN and RNN in the developed model were examined. Extracting spatial–temporal features of rainfall distribution data using CNN and RNN was shown to be an effective means of predicting the water level with high accuracy. In particular, RNN, which extracts information on past rainfall histories, was indispensable in the prediction of river water levels.

ACKNOWLEDGEMENTS

Graduates of our laboratory, Mr. Ishizaki, M. and Mr. Fukuoka, M., cooperated in this study. We thank DIAS (Data Integration and Analysis System) for distributing the XRAIN data.

REFERENCES

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481-2495.

Graves, A. and Jaitly. N. (2014). Towards End-to-End Speech Recognition with Recurrent Neural Networks, *Proceedings of the 31st International Conference on Machine Learning*, 32(2):1764-1772.

Hitokoto, M., M. Sakuraba, and Y. Sei (2017). Development of the real-time river stage prediction method using deep learning, *Journal of JSCE*, 5(1):422-429.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory, *Neural Computation*, 9(8):1735-1780.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, 25(2):1097-1105.

Mcmillan, H., Freer, J., Pappenberger, F., Krueger, T., and Clark, M. (2010). Impacts of uncertain river flow data on rainfall-runoff model calibration and discharge predictions, *Hydrological Processes*, 24(10):1270-1284.

Mikolov, T., Karafiát, M. Burget, L., Cernocký , J., and Khudanpur, S. (2010). Recurrent neural network based language model, *Interspeech-2010*:1045-1048.

Moradkhani, H. and Hsu, K., Gupta, H., and Sorooshian, S. (2005). Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter, *Water Resources Research*, 41(5):W05012.

Nakatani, Y., Ishizaki, M., and Nishida, S. (2017). Estimation of water quality variation in a tidal river by applying deep learning models, *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)*, 73(4):I_1141-I_1146.