Investigation of Long Short-term Memory Model Applied for Imputation of Missing Discharge Data

WEILISI

June 2022

Acknowledgements

I am deeply grateful to my supervisor, Associate Professor Toshiharu Kojima for his guidance and assistance for this work. I would also like to thank Professor Seirou Shinoda and Assistant Professor Keisuke Ohashi for their discussions and advices for my research. My sincere thanks also go to Professor Ichiro Tamagawa for serving on my dissertation committee.

My friend Xiaoyu Wang, Tonghuan Yu and Baiqiang Zhang are acknowledged for their academic suggestions that helped me a lot on the writing of the papers. All the members of Laboratory of Environmental Hydrology are appreciated for their help and suggestions on my life in Japan.

Lastly, many thanks to my parents and wife for supporting me in both life and study.

Abstract

Missing data in hydrological field is still an inevitable problem at present day. Solving this kind of problem is an important task for river management. Upgrading the observation devices could only lower the missing rate of data. There will still be possibility of missing value. At the same time, budget and installation environment should also be considered for upgrading the devices. Thus, complement of missing data is another way to solve the problem. At the present stage, there are already many methods to complement missing hydrological data which were proved practical. On one hand, short period of data gaps can be complemented from data around the gaps. On the other hand, long period of data gaps could be complemented by tank model or data of neighboring observation points. However, these kinds of traditional method have their limits such as low accuracy at extreme hydrological conditions, large amount of data conversion and lots of time consumption.

Deep learning technology has developed quickly in recent decades and has advantages than the traditional methods on the prediction of hydrological data. There are many applications on runoff prediction and flood forecasting. However, deep learning is not applied commonly in the application of missing data complement at present because in the case of a short missing period, interpolation using observation data before and after the missing period is useful, or complement using observation data from different points in upper stream and lower stream is relatively accurate. These methods are solutions of missing data problem to a certain extent due to their own limits. If there is a long missing period, or the observation points are far from each other, deep learning might be a better choice.

In this study, long short-term memory (LSTM) is applied to complement missing discharge data of the Daihachiga River and verified. Stochastic gradient descent (SGD) is generally used as the optimization algorithm for deep learning. SGD can obtain good learning results quickly, but the results are slightly different for each learning by its randomized algorithm. In the first part, learning was performed 1000 times and it was recognized that the difference between each learning result is non-ignorable.

The ensemble average, which obtained by more than 20 learning results, indicates 0.904 in 5th percentile of Nash-Sutcliffe efficiency which is higher than the accuracy of linear regression analysis. The method to reduce the variation of learning results and obtain sufficient accuracy is proposed. In the second part, the hyperparameters of the LSTM were tuned to investigate their influence on the model performance. The result indicated that tuning of hyperparameters is required and it can influence the performance of training to a certain extent. In the analysis of the second part, it was proved that the optimal hyperparameter combination could improve the performance of the model several times for missing discharge data imputation. The hyperparameter setting of this study could be a reference for further research on the relevant area.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Water level and discharge of river are the most basic and important hydrological information for river construction and management, and there are many measurement data. In particular, river water level is easy to measure steadily against the flow rate and is measured at many points. Integrity of such hydrological data plays an important role in river management. Especially in the case of disaster prevention and mitigation, missing data may cause mistake in decision-making and have serious consequences. Although information on river disaster prevention [1], such as rainfall and water level, is distributed by telemeter, it was reported that the percentage of missing/undetected data and abnormal values by observation type at national and prefectural sites in 2012 is as follows: water level 63.0%, precipitation 8.9%, water quality 11.3% and water level data has a high rate of occurrence on missing, not received, or abnormal values. It could be said that complementing missing water level data is an important task for river management. For rainfall, the most common cause of missing/unreceived data and abnormal values is abnormal communication path (about 40%), while for water level, it is reported that more than 70% of all missing data are due to the failure of the measurement device or the water level dropping below the lower limit of measurement of the water level gauge [2]. Applying better hardware will improve efficiency of data acquisition and transfer and prevent data gaps to a certain extent [3]. However, improving devices could only lower the missing rate of data. There will still be possibility of missing value. At the same time, budget and installation environment should also be considered for upgrading the devices.

In the case of inevitability of missing data at the present stage, complementing is necessary and feasible. There are already many methods to complement missing hydrological data which were proved practical. On one hand, short period of data gaps can be complemented from data around the gaps. On the other hand, long period of data gaps could be complemented by tank model or data of neighboring observation points. Aburatani et al. [4] compared the interpolation methods of missing values by linear interpolation, 6-point scheme, and tank model using the observed values near the missing interval. They reported that the tank model can generally supplement missing values in all situations, but it requires a lot of

effort and time, and that the linear interpolation and 6-point scheme may be more suitable than the tank model because interpolation is difficult during floods and less effort is required during normal water level. Besides, Kojiri et al. [5] applied pattern classification and fuzzy inference on the complement of missing values. Runoff model is another option, but plenty of data and data conversion are needed before the calculation. Its demerit is similar with the tank model mentioned above.

Deep learning and Artificial Neural Networks (ANNs) have made remarkable progress in recent years, and there are many applications in the field of river technology. A three-layer hierarchical neural network with input layer, intermediate layer, and output layer, also known as shallow neural network, has been commonly used in the past [6,7]. However, in recent years, as more computational resources have become available, Deep Neural Networks (DNN) with multiple intermediate layers have come into use. Wada et al. [8] proposed a Genetic Programming Neural Network (GPNN) that determines the structure of the input layer using a genetic algorithm, as opposed to a DNN with five intermediate layers, and performed long-term flow forecasting. Hitokoto et al. [9,10] and Fusamae and Shimamoto [11] have also conducted flood forecasting using a four-layer DNN with two intermediate layers. Besides, there are various applications, such as the application to the prediction of water level in a drainage pumping station by Kimura et al. [12] and the application to the long-term prediction of dam inflow by Tamura et al. [13].

In the current so-called third AI boom, many ANNs from relatively simple hierarchical structures to with more complex structures are being used. On one hand, Convolutional Neural Networks (CNN) have two-dimensional image data as input data, a convolutional layer that corresponds to the convolution of the image, and a pooling layer that processes the features extracted by the convolutional layer so that they are not affected by translation. It has many applications in the fields of image recognition, such as face recognition and automatic driving, and video recognition. CNNs using 2D data have few applications in the field of rivers where time series data is used, but it is possible to use 1D data by converting it to 2D. Kimura et al. [14] used 16 variables such as rainfall

and water level as input data, and converted them into 2D data with observation points arranged from upstream to downstream as the vertical axis and time series as the horizontal axis, and applied them to CNN for flood forecasting. On the other hand, Recurrent Neural Network (RNN) has a structure that temporarily stores the state of the intermediate and output layers and uses them for the next input in order to analyze time-series data, and is often used for machine translation and speech recognition as an ANN suitable for using time-series data. In the field of rivers, for example, Teranishi and Shidawara [15] combined the Jordan type, which uses the state of the output layer to the intermediate layer, and the Elman type, which also uses the state of the intermediate layer to the intermediate layer, and applied them to runoff prediction during floods; Taniguchi et al. [16] applied them to dam inflow prediction; RNNs with Elman-type networks applied to both the intermediate and output layers were applied to real time sewer water level prediction by Chiang et al. [17]; and the prediction accuracy of stream flow by hierarchical neural networks and RNNs was compared by Sahoo et al. [18].

However, simple RNNs such as the Elman and Jordan types have a problem that is impossible to take very old information into account in the learning due to the vanishing gradient problem. In order to alleviate this vanishing gradient problem, RNN models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been proposed, and there are relatively many applications of LSTM in the field of hydrology. Especially in the applications of runoff prediction and flood forecasting, LSTM is one of the commonly used networks [19]. Hu et al. [20] got conclusion that LSTM model had better simulation performance and more stable than ANN model on simulation of rainfall-runoff process. Lee et al. [21] performed runoff simulations of river using LSTM model and a physics-based model and designated that LSTM model is more applicable. Lee et al. [22] and Xiang et al. [23] presented an application of LSTM model on simulation of hydroclimatological variables and estimation of hourly rainfall-runoff, respectively, and pointed out that the LSTM model had better performance than traditional models. Fan et al. [24] applied LSTM on runoff modeling of a river basin and compared performance of the LSTM with ANN and Soil & Water Assessment Tool

(SWAT). It is indicated that the performance of LSTM is better than others. Bai et al. [25] compared robustness of LSTM model with two hydrologic models in prediction of runoff in changing climatic conditions and considered that the LSTM model is a preferred tool for runoff simulation with sufficient calibration data. Kratzert et al. [26] studied the ability of LSTM to predict discharge of a catchment and discussed potential of LSTM in application of hydrological modeling. Sudriani et al. [27] analyzed daily discharge data by LSTM and held that the relative error was acceptable. Ding et al. [28], Le et al. [29], Li et al. [30], and Zhang et al. [31] worked on the water table depth prediction and flood forecasting of LSTM by setting hydrologic data as input data. And prediction result of the LSTM model was effective. Sahoo et al. [32] and Qin et al. [33] explored the suitability of LSTM for prediction of low-flow time series and hydrological time series, respectively. These two studies showed the feasibility of LSTM model for predicting discharge and hydrological time series. Additionally, the potential of LSTM in forecasting of evapotranspiration was also explored and the fact was known that the performance of the model can be affected by different factors. Granata and Di Nunno [34] deployed LSTM and nonlinear autoregressive network with exogenous inputs (NARX) for prediction of evapotranspiration. The result indicated that each model has its own advantages in different climatic conditions since the model performance can be affected by local climatic conditions significantly. Chen et al. [35] compared the performance of LSTM with several other models in estimation of daily reference evapotranspiration. The model performance was influenced by the type of available features. Ferreira and da Cunha [36] investigated the potential of deep learning models, machine learning models, and a combined model in forecasting of daily reference evapotranspiration in local and regional scenarios. Even if the model performance varied with different input data combinations, the combined model, which consisted of LSTM and a CNN, had the best accuracy in both local and regional scenarios. The various algorithms used in deep learning are packaged and modularized in programming languages such as Python and MATLAB, making it relatively easy for even beginners to adopt them. However, in order to obtain good results, trial-and-error processes are required, such as the

selection of appropriate models and fine tuning, and it is necessary to continue to conduct a large number of analyses for each application target and accumulate a variety of knowledge.

However, tuning of hyperparameters influences the performance of an LSTM model. Exploring optimal hyperparameters for an LSTM model is already a study objective in applications of LSTM models in fields other than hydrology, such as sequence labelling [37], network attack detection [38], stock market prediction [39], highway traffic prediction [40], etc. In the case of hydrology, the tuning of hyperparameters is a necessary step before the application of the model in much another research [28,30,41–44]; however, no sufficient knowledge has been obtained.

Comparing missing data complement by deep learning with flood forecasting or long-term water level prediction, there are only few applications because in the case of a short missing period, interpolation using observation data before and after the missing period is useful, as in the case of Aburatani et al. [4] , or complement using observation data from different points in upper stream and lower stream is relatively accurate. But the potential of deep learning is being explored in those applications. Dastorani et al. [45] predicted missing flow data by ANN and adaptive neuro-fuzzy inference system (ANFIS). It is presented that ANFIS technique has superior ability and ANN is an efficient method to predict missing flow data. Mispan et al. [46] tried prediction of missing stream flow data with ANN model and illustrated that ANN can predict the missing data well. There are some differences in the application of deep learning between water level/flow rate prediction and missing data complement in terms of the following: (1) flood prediction is for several hours in the future, while missing data complement depends on the missing period, and thus is for one to several days in the future; (2) missing data complement can use future observation data; (3) missing data complement can use observation data at the time of estimation, as long as it is at a different location from the one to be estimated. There are two types of complement: interpolation, which uses surrounding values to fill in the gaps, and imputation, which means to make something complete by filling in the missing parts, as used by Kojiri et al. [5].

The purpose of this study is to obtain a complete time series of river discharge data by complementing the missing discharge data with the flow data from other locations. An LSTM which is suitable for processing time series information, is applied and its accuracy is verified. The hyperparameters of an LSTM model was tuned to investigate the influence on the model performance and an attempt was made to obtain a more suitable hyperparameter combination in the imputation of missing data using an LSTM model. A suitable hyperparameter combination of LSTM will improve the performance of the model on missing discharge data imputation and it will have reference value in hydrological research.

Chapter 2

Investigation of Missing River Discharge Data Imputation Method Using Deep Learning

2.1 Target River Basin, Data and Imputation by Linear Regression Analysis

2.1.1 Target River Basin and Used Data

The Daihachiga River basin of the Miya River system in Takayama City, Gifu Prefecture, is the target river basin. In the Daihachiga River basin, water level and discharge measurements have been conducted at hourly intervals since 1986 at two points, Sanpukuji (catchment area: 60 km^2) and Shioyabashi (catchment area: 41 km^2), by the Miya River Upper Reaches Development and Construction Work Office of Gifu Prefecture. In this study, the observed discharge data of 2008, which has no missing data in both locations, will be used for training and the observed discharge data of 2009 will be used for validation. The target river basin is shown in Figure 2-1.

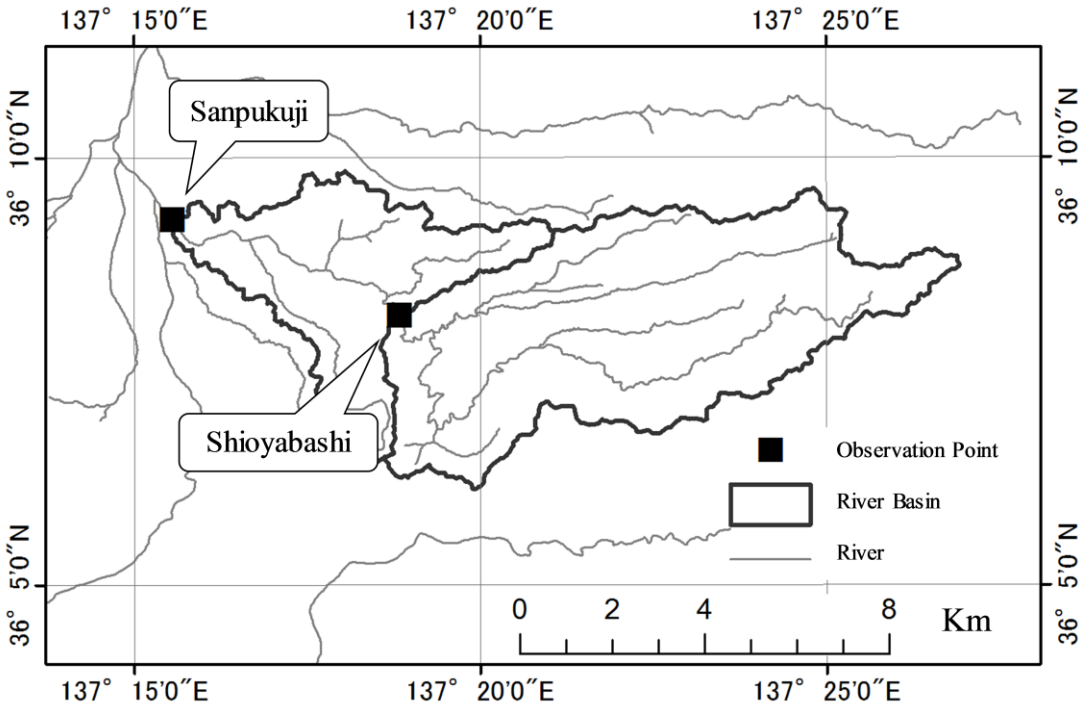

Figure 2-1. Target river basin.

2.1.2 Imputation by Linear Regression Analysis

Figure 2-2 shows a scatter plot of the discharge of Sanpukuji Q_{san} (m^3/s) and the discharge of Shioyabashi Q_{shio} (m^3/s) in 2008. The distance

between the two points is 6.4 km, which is relatively short, and the correlation coefficient between the discharge of two points is 0.915, which is very high. On the other hand, there is some dispersion in the peak period, and differences are observed among rainfall events. The following equation is obtained from the scatter plot as a linear regression line.

$$Q_{shio} = 0.7280Q_{san} + 0.1657 \quad (1)$$

Using equation (1), the hydrograph shown in Figure 2-3 was obtained by estimating Q_{shio} from Q_{san} in 2009. In the figure, the horizontal axis is DOY (day of year) and Q_{est} is the estimated result of Q_{shio}. When the Nash-Sutcliffe model efficiency coefficient (NSE) was calculated, it was found that NSE = 0.903 which is highly accurate for runoff prediction. However, there are underestimations in the peaks of discharge and some underestimations in the reduced discharge and normal water conditions. In this study, it was investigated whether these underestimations can be corrected by deep learning.

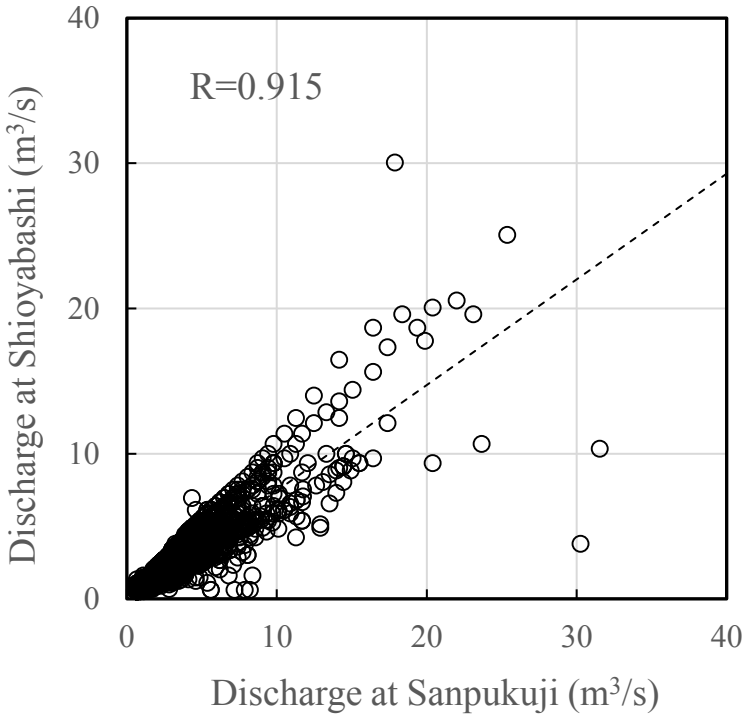

Figure 2-2. Relationship between discharge of Sanpukuji and Shioyabashi.

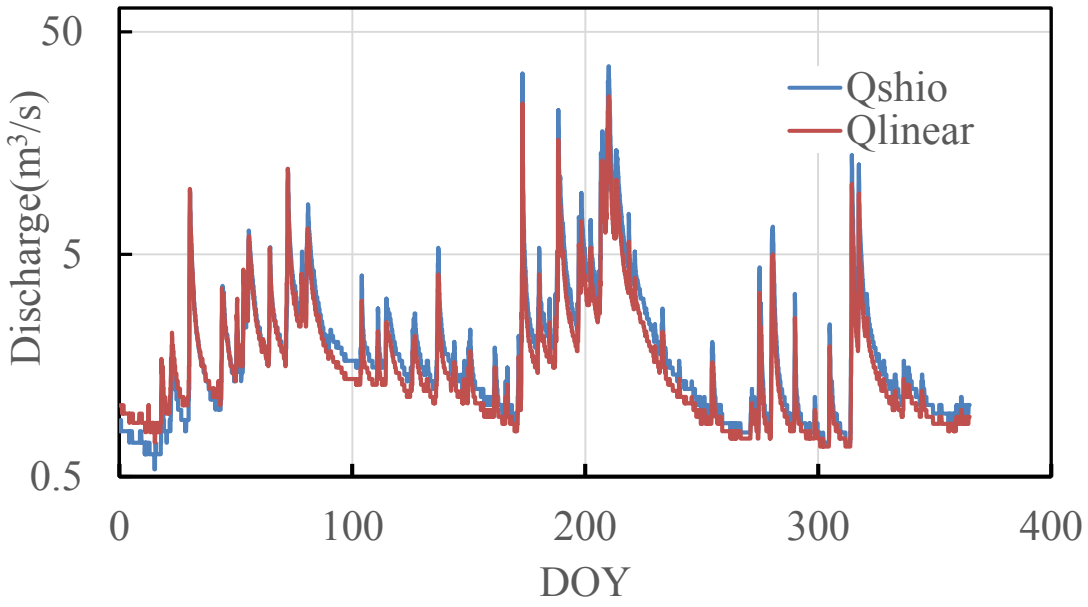

Figure 2-3. Comparison of the discharge estimated by the linear regression equation with the observed value.

2.2 LSTM Model

2.2.1 LSTM Model

The structure of the LSTM model used in this study is shown in Figure 2-4. In the LSTM, the internal state C_t of the LSTM unit at time t and the output h_t from the LSTM unit are transmitted to the LSTM at the next time $t+1$. In the LSTM unit, the input values x_t at time t and h_{t-1}, C_{t-1} are used to obtain the output h_t at time t. In the output layer unit, the output y_t from the output layer is obtained from the output h_t from the LSTM. The above procedure is the same as the standard RNN model. In the case of LSTM, x_t, h_{t-1} and C_{t-1} will be passed through the input gate, output gate and forget gate in the LSTM unit and the vanishing gradient problem can be solved [47,48]. In this study, the number of units in the input layer is 24, the number of units in the output layer is 5, and the number of units in the hidden layer (LSTM layer) is 50, as shown in Figure 2-4. The model was implemented using Python 3.7.3, Keras 2.2.4, and Tensorflow 1.13.1.

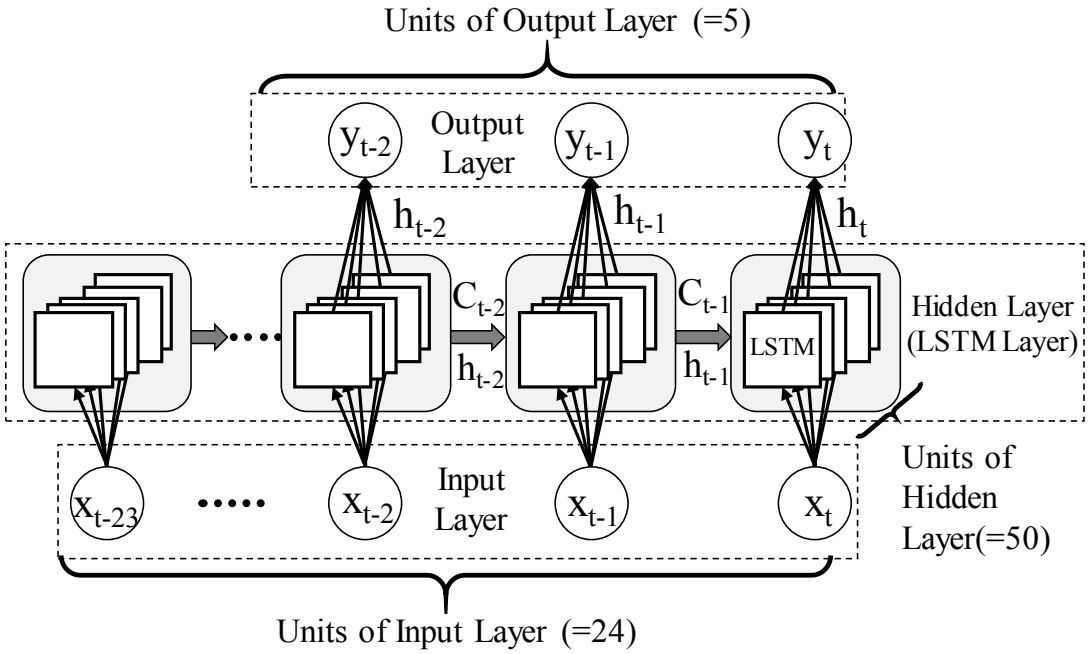

Figure 2-4. Structure of the LSTM model used in this study.

The activation function, dropout rate, and other settings inside the LSTM were left as default (i.e., tanh for activation function and 0 for dropout rate). The activation function of the output layer is elu (Exponential Linear Unit), and Adam is used as the optimizer. Mean_squared_error was used as the loss function. The training data was randomly divided into 90% training data and 10% validation data during the training. The maximum number of training times (number of epochs) were set as 200, and to prevent overtraining, EarlyStopping was set to stop after 20 times of training after the error evaluation (val_loss) stopped decreasing using the validation data during training.

2.2.2 Input Data and Output Data

Assuming that *t* is the time of the output value to be estimated, the output layer outputs the discharge of Shioyabashi from *t* to *t-4*. The input layer inputs the discharge of Sanpukuji from *t* to *t-23* and the discharge of Shioyabashi from *t-24* to *t-47*. All time intervals are 1 hour. Table 2-1 shows the types of input data and the number of variables for each case.

Table 2-1. Type of input data and number of variables

No.	Input Variables	Number of Input Variables
Case1	$Q_{shio} + Q_{san}$	2
Case2	Q_{shio}	1
Case3	Q_{san}	1

2.3 Results and Discussion

2.3.1 Investigation of Dispersion in the Training Results

Because Stochastic Gradient Descent (SGD), a method that incorporates randomness in the initial values and the learning process, is generally used for ANN optimization, the model coefficients obtained through trainings are slightly different for each training. Figure 2-5 shows three examples of training results for Case 1. These three cases use the same model structure, the same settings, and the same training data, but each time the model is restarted from the initial settings, and the training results are slightly different. Training results 1 and 2 in Figure 2-5(a) and Figure 2-5(b) show NSE of 0.932 and 0.902, respectively, which are equivalent to or better than the estimation results by the linear regression model, in the accuracy evaluation using the validation data, and the underestimation of peak discharge, discharge reduction and normal discharge is improved. However, in the training result 3 shown in Figure 2-5(c), the NSE is 0.463, which shows poor reproducibility especially in the peak discharge, indicating that there is dispersion in the training results. The evaluation values of the loss function by the training data (loss) and the loss function by the validation data (val_loss) of the training results 1~3 are almost the same as shown in Table 2-2, and it is difficult to check the applicability and versatility other than the training data at the time of training, although it is checked by val_loss. To see how the dispersion of the training results occurred, 1000 independent trainings were carried out, and the loss, val_loss of the trainings were compared with the NSE which used the training data and the NSE which used validation data (Figure 2-6). The correlation between the NSE using the training data and the loss, which is the error evaluation value from the training data that accounts for 90% of the data that used for training, is relatively high at R=-0.683 (Figure

2-6(a)). However, the other correlations (Figure 2-6(b), (c), and (d)) are low, especially the correlations between the NSE using the validation data and loss, val_loss (Figure 2-6(c) and (d)) are -0.0166 and 0.0369, respectively, which is almost uncorrelated, indicating that even if the error evaluation results during the training were good, the accuracy during the application could often be very poor.

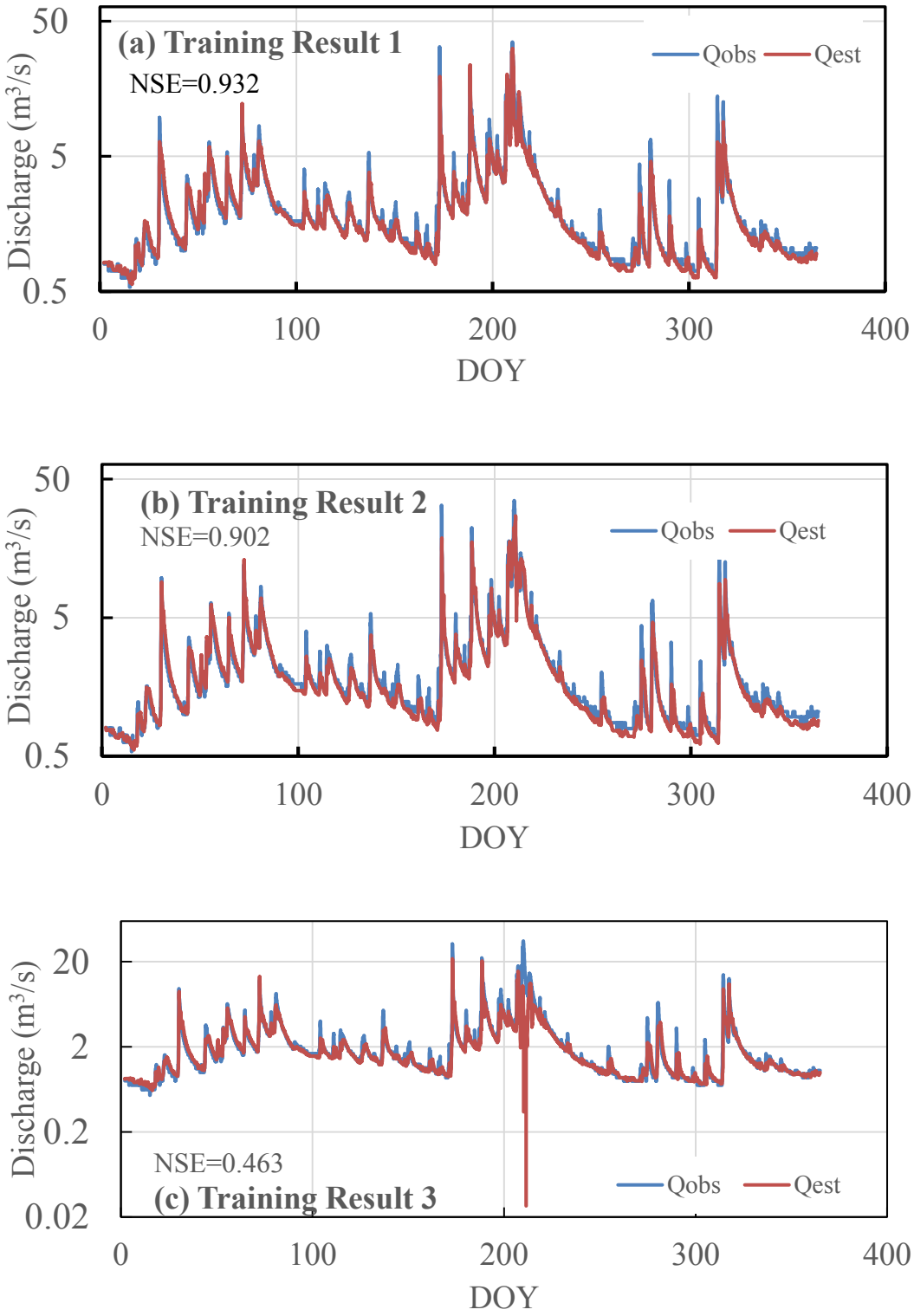

Figure 2-5. Examples of training results for Case 1.

Table 2-2. Accuracy evaluation for each training result.

No.	NSE (validation data)	loss	val_loss
1	0.932	7.88×10^{-5}	2.40×10^{-5}
2	0.902	8.70×10^{-5}	1.96×10^{-5}
3	0.463	6.80×10^{-5}	2.15×10^{-5}

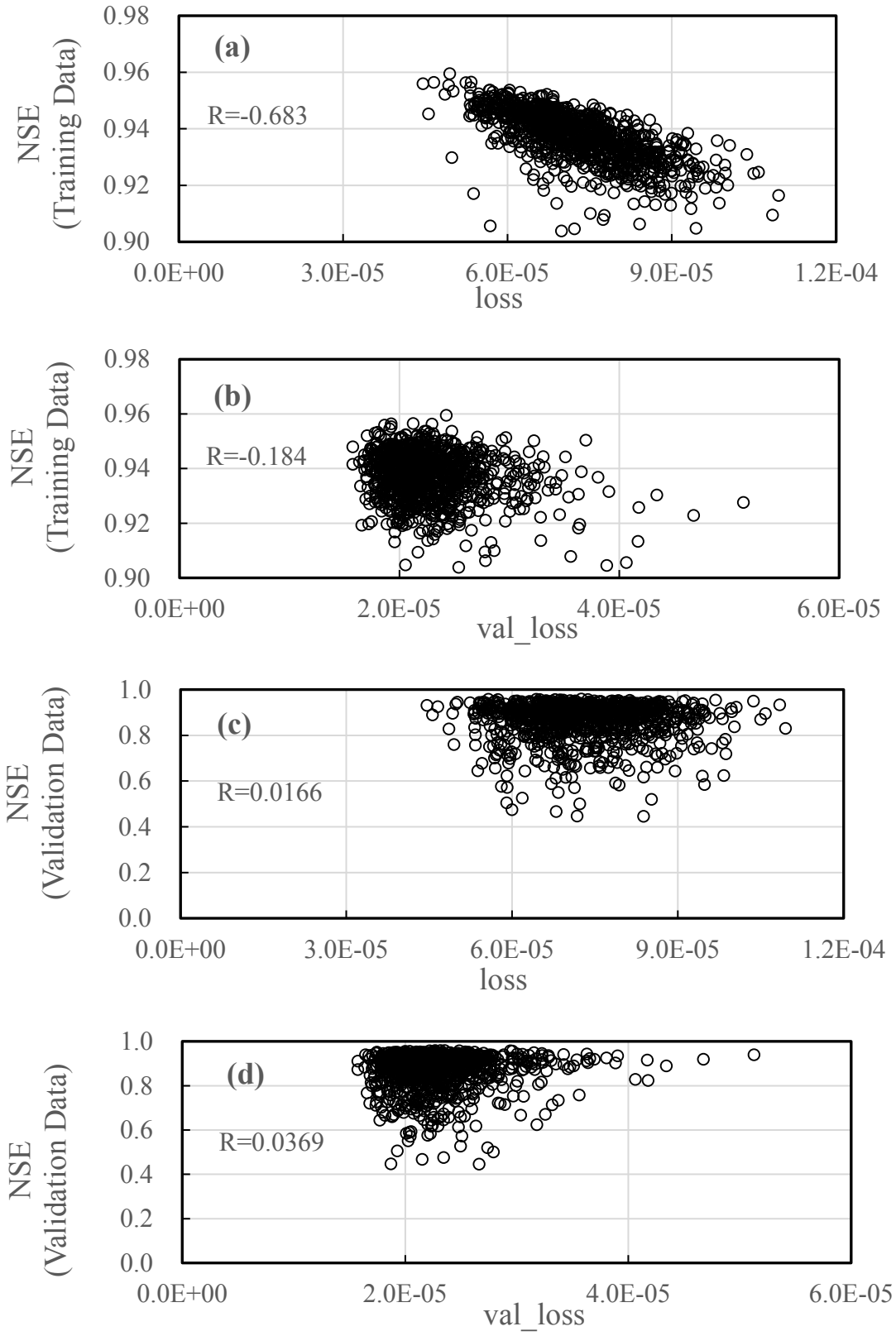

Figure 2-6. Comparison of the accuracy ratings of training results.

Next, the hydrograph was estimated by N randomly selected training results from the 1000 training results and the accuracy of the ensemble average of the N hydrographs was evaluated. The number of average hydrographs to be calculated was set to 40, the mean and standard deviation of the NSE of the ensemble averages from N=1 to 50 were calculated, and the change of 5th percentile was examined (Figure 2-7). The mean of the NSE exceeded the accuracy of the linear regression equation of 0.903 for the average hydrographs with $N \geqslant 2$, but the accuracy of the 5th percentile was 0.829, which is still below the accuracy of the linear regression equation in many cases. For the average NSE, the accuracy of the linear regression equation of 0.903 is exceeded for the average hydrograph of $N \geqslant 2$. The NSE of 5th percentile increased rapidly from 0.768 at N=1 to 0.903 at N=10, and after a slight increase and decrease, the accuracy always exceeded NSE=0.904 for $N \geqslant 20$. This shows that in Case 1, the average hydrograph of more than 20 training results was calculated to suppress the dispersion of the training results and almost surpass the accuracy of the linear regression equation.

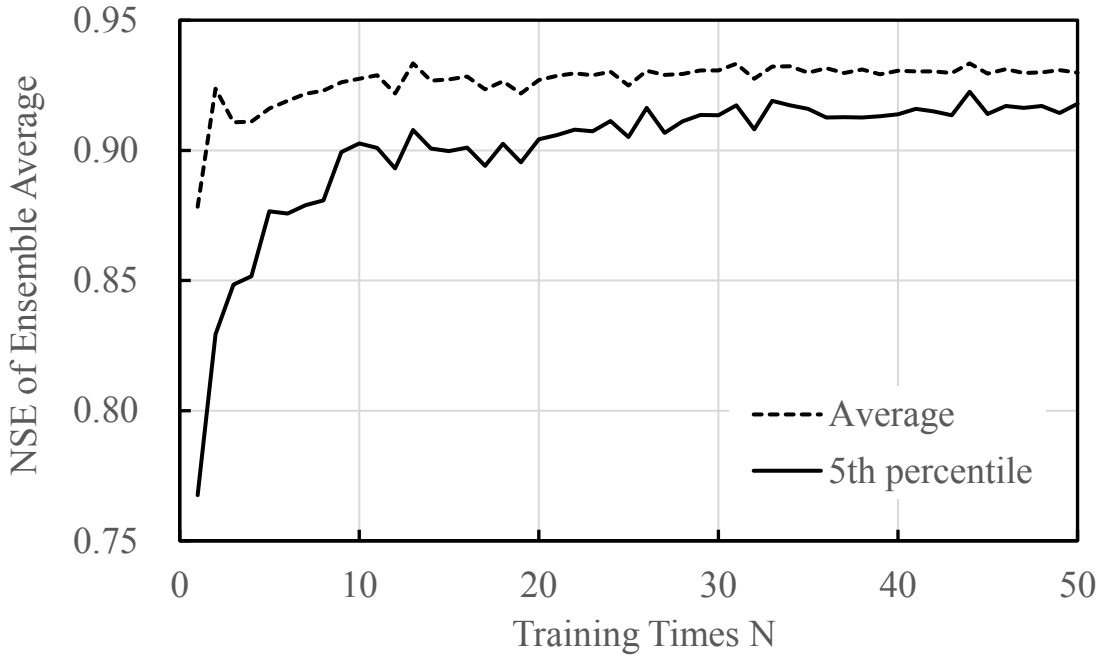

Figure 2-7. Relationship between training times and NSE (Case 1).

2.3.2 Consideration of Differences in Input Data

In Case 2 and Case 3, as in Case 1, the relationship between the number of training times and the NSE of the ensemble average was evaluated. In Case 2, the accuracy was very low, with an average value of 0.530. In the 5th percentile, the accuracy improved as N increased and became almost constant at $N \geqslant 30$, however the value was about 0.52, which was not enough to obtain sufficient accuracy (Figure 2-8). Case 3 is different from Case 1, although the NSE of the 5th percentile was 0.865 for N=1, showing high accuracy even if there was few number of training times, it was almost constant at about 0.901 for $N \geqslant 15$. On average, the accuracy was always higher than 0.905, indicating that the accuracy exceeded the linear regression equation with a probability of 50% (Figure 2-9). As a result, Case 1, which uses both the time series data of the target of missing value estimation and the time series data of the neighboring observation points, has the highest estimation accuracy, and shows a higher accuracy than the linear regression equation to be compared. The reason why Case 1 had the best accuracy is that Case 2 uses only the time series data of Shioyabashi, which is the target of the estimation, and it is difficult to estimate whether the discharge will increase or decrease in the next 24 hours; in Case 3, only the time series data of the neighboring observation point, Sanpukuji, is used, and no correction based on the previous Shioyabashi is made.

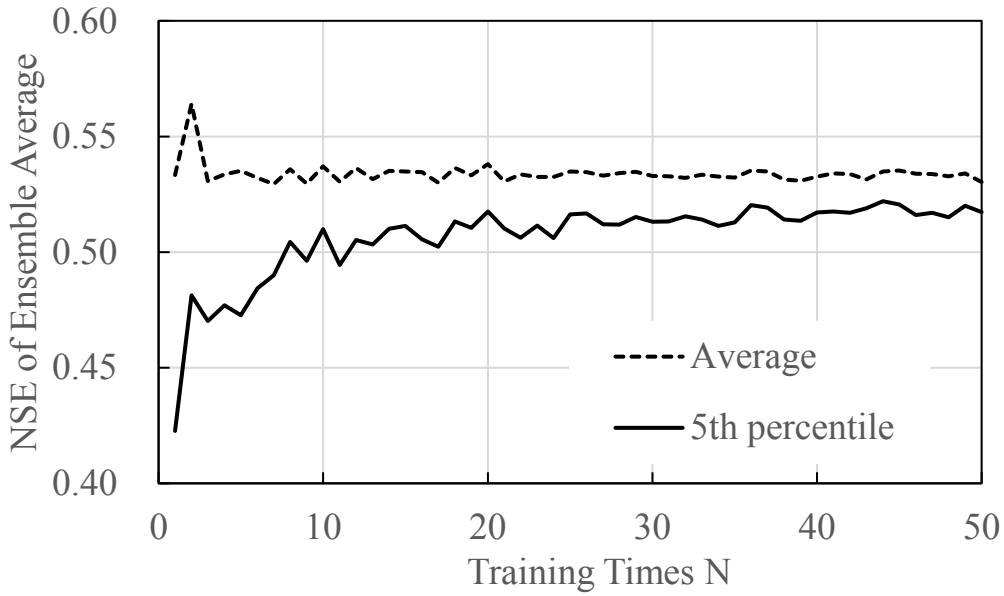

Figure 2-8. Relationship between training times and NSE (Case 2).

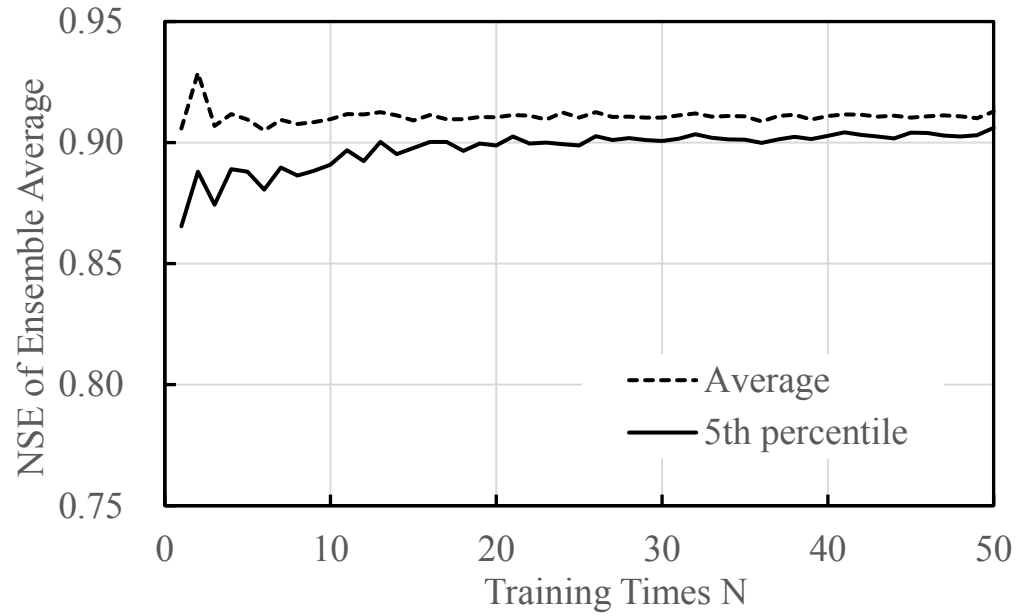

Figure 2-9. Relationship between training times and NSE (Case 3).

2.4 Conclusion

In this study, a completion method for missing discharge values was investigated by LSTM, which belongs to recurrent neural networks that are

good at analyzing time series data. The SGD with randomness, which has been used for neural network optimization in recent years, can obtain good training results on average at high speed, however the obtained training results are dispersed. In general, such dispersion is so small as to be negligible, however, depending on the application target, the accuracy during training may be high but may not be sufficient when applied to other data. This phenomenon is called overtraining, and since ANN requires a very long computation time for each training, it is common to construct a highly versatile ANN by examining the training data, constructing the model, and tuning the hyperparameters. On the other hand, since the time required for one training session is very short (1~10 minutes) for the application target of this study, the implementation of a large amount of trainings and the dispersion of the training results were investigated. After 1000 times of training, the results suggest that the dispersion is not negligible. However, the average hydrographs of more than 20 times of training results almost always showed better accuracy than the linear regression equation to be compared, suggesting a method that is able to obtain sufficient estimation accuracy without a lot of detailed trial and error such as tuning of hyperparameters. Besides, for the application of the imputation method to the missing discharge data, it was shown that it is very effective to use the discharge data of another neighboring observation point as input data, in addition to the discharge data except for missing period at the target point for precise estimation, to improve the estimation accuracy.

Chapter 3

Investigation of Hyperparameter Setting of Long Short-term Memory Model Applied for Imputation of Missing Discharge Data of the Daihachiga River

3.1 Materials and Methods

3.1.1 Study Area

The study area of the present work is the Daihachiga River Basin (Figure 3-1). Although it is the same area mentioned in Chapter 2, it will be reiterated here. The Daihachiga River belongs to the Jinzu River system. It starts from Hikagedaira Mountain, which is located in the east of Takayama City, Gifu Prefecture, Japan. The river flows from the east to the west, and merges into the Miya River in the urban area of Takayama. The river basin has about 1800 mm of mean annual precipitation, 10.9 ℃ of mean annual air temperature, and 60.4 km^2 of catchment area [49,50].

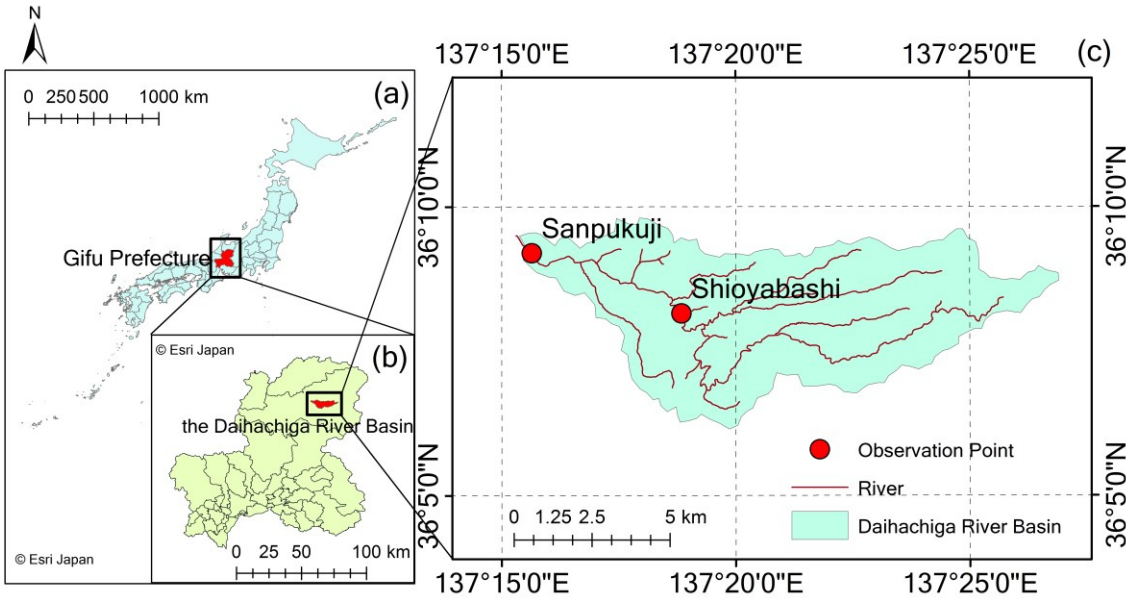

Figure 3-1. The Daihachiga River Basin and observation points. (a) Location of Gifu Prefecture; (b) location of the Daihachiga River Basin; (c) observation points of the Daihachiga River.

3.1.2 Data

Discharge data of the Daihachiga River were used in this study. The data were observed at one-hour intervals by the Gifu Prefecture at the observation points of Shioyabashi (36°8'8"N, 137°18'50"E, 617 m) and Sanpukuji (36°9'12"N, 137°15'38"E, 563 m) (Figure 3-1). The data of 2008 and 2009 were used as training and validation data, respectively. In

addition, precipitation and air temperature data at one-hour intervals observed by the Japan Meteorological Agency in 2008 and 2009, Takayama (36°09'18"N, 137°15'11"E, 561 m), were used during the training of the discharge data.

3.1.3 Hyperparameters of LSTM Model

The structure of LSTM model used in this study is almost same as the one mentioned in chapter 2 (Figure 2-4). The only difference is that the hyperparameter of the model in this study is not fixed value. The hyperparameters tuned for training are shown below:

- In: number and type of input variables.
- $Back_{ts}$: backtracked time steps of data used for the training.
- Hid: number of units of hidden layer.
- Drp: dropout.
- Drp_r: recurrent dropout.

The purpose of this study is to impute the missing values. Therefore, it is possible to use past or future data of the observation point for which the missing value must be estimated as input values. Because LSTM is a structure that propagates information from the past to the future, future information is not as effective as present information as input data. Therefore, in this study, the past data $Back_{ts}$ steps before will be used as training data. When the data to be estimated are used as input data for training, considering the existence of missing data, the data from t-$Back_{ts}$ to t-$Back_{ts}$-24 are used to estimate the data at time t. This means that if there is a $Back_{ts}$ steps gap of missing data, this part can be estimated by the data before the gap. Thus, the hyperparameter $Back_{ts}$ only has a value when the estimated data were also used as input data. For other input data than the estimated point, such as temperature and precipitation, data from time t to t-24 are used to estimate time t.

The LSTM was actualized by Python 3.7.4 with the Keras 2.3.1 and Numpy 1.18.1 libraries. The activation function was an exponential linear unit (ELU) and the optimizer was Adam.

3.1.4 Hyperparameter Settings and Data Training

LSTM training was carried out by 90 kinds of hyperparameter combination (Table A1) settings to figure out the best one. The discharge data of the Daihachiga River, and the air temperature and precipitation data of Takayama were used as input variables. Several specified values were assigned for each hyperparameter in the training (Table 3-1). $Back_{ts}$ = 24 and 168 assume the missing period of 1 day and 7 days, respectively. In each training period, values of dropout and recurrent dropout were assigned identically. The hyperparameter values were assigned by trial-and-error approach in those 90 trainings. The estimated data were the discharge of Shioyabashi. Table 3-2 shows the input variable types for each case. For example, in the setting of $Back_{ts}$ = 24, Case 1 takes as input data two variables: the discharge data from t-$Back_{ts}$ to t-$Back_{ts}$-24 at the Shioyabashi, and the precipitation at t to t-24. Case 2 takes as input data two variables: the discharge data from t-$Back_{ts}$ to t-$Back_{ts}$-24 at the Shioyabashi, and the discharge data from t to t-24 at Sanpukuji. Case 5 takes one variable as input data: the discharge data from t-$Back_{ts}$ to t-$Back_{ts}$-24 at the Shioyabashi. Additionally, Case 9 takes as input data one variable: the discharge data from t to t-24 at the Sanpukuji.

Table 3-1. Assigned values of hyperparameters.

Hyperparameter	Value1	Value2	Value3	Value4
$Back_{ts}$	24	168	0	
Hid	20	50	100	200
Drp	0	0.01	0.05	0.1
Drp_r	0	0.01	0.05	0.1

Table 3-2. Cases of input variables.

	Type of Input Variables	Number of Input Variables
Case1	$Q_{shio} + P$	2
Case2	$Q_{shio} + Q_{san}$	2
Case3	$Q_{shio} + Q_{san} + P$	3
Case4	$Q_{shio} + Q_{san} + P + T$	4
Case5	Q_{shio}	1
Case6	$Q_{shio} + T$	2
Case7	P	1
Case8	T	1
Case9	Q_{san}	1
Case10	$Q_{san} + P$	2
Case11	$Q_{san} + T$	2

P: precipitation. Q_{san}: discharge volume of Sanpukuji
Q_{shio}: discharge volume of Shioyabashi T: air temperature

The evaluation method of chapter 2 was referenced and improved advisably in this study. Because the result of each training of Stochastic Gradient Decent (SGD) is different, which means the accuracy of each training is different, the ensemble average of multiple training results was calculated to cancel such differences. For each combination of hyperparameters, the training was repeated 500 times. From the results of 500 trainings, N ensemble members of them were picked out randomly and the ensemble average was calculated. The Nash–Sutcliffe model efficiency coefficients (NSE)[51] of these average values were calculated for the evaluation. Additionally, the evaluation metrics commonly used on deep learning, such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were regarded as the potential options to evaluate the model accuracy of this study. The N varies from 1 to 50 and for each N, the random pick out was carried out 40 times. The average and standard deviation of NSE of these 40 ensemble average values were calculated and the variation of 5th percentile (P_5) on N = 1–50 was evaluated. Even if the result was dispersed, P_5 can indicate that 95% of the results were better than it.

3.1.5 Traditional Methods

The linear regression equation was used as the traditional imputation method compared with the new method using deep learning, which was proposed in this study. When y_t is the discharge data observed on Shioyabashi at time t, and x_t is the discharge data observed on Sanpukuji at time t, the linear regression model is defined as the follows equation (2):

$$y_t = ax_t + b \qquad (2)$$

where a and b are the regression parameters. As mentioned in chapter 2, a and b have been estimated with the data of 2008, as shown in Figure 2-2. The x axis is the discharge data at Sanpukuji (x_t), and the y axis is the discharge data at Shioyabashi (y_t), where the correlation coefficient R was 0.915. The dotted line is the linear regression model. The accuracy, which was evaluated with the discharge data in 2009, was 0.903 in NSE.

A tank model optimized by Shuffled Complex Evolution (SCE-UA) method was also considered as the comparison target of the deep learning. In this model, the precipitation of Takayama in 2009 was used as input and the discharge height was estimated. The equation of the model is shown as follows equation (3):

$$\Sigma Q = \Sigma R - \Sigma E \qquad (3)$$

where Q is the discharge height (mm/h), R is the precipitation and E is the evaporation. The discharge coefficient was set as 0.9. The evaporation was set as 0 since it was complicated and hard to grasp the exact amount of evaporation in this case. However, the R of the tank model was 0.888 (Figure 3-2), and the NSE was 0.771, which is lower than the linear regression model. The aim of this study is to propose a new method to obtain an improved accuracy. Thus, the higher accuracy of the linear regression model was chosen as the goal to be exceeded by the new method.

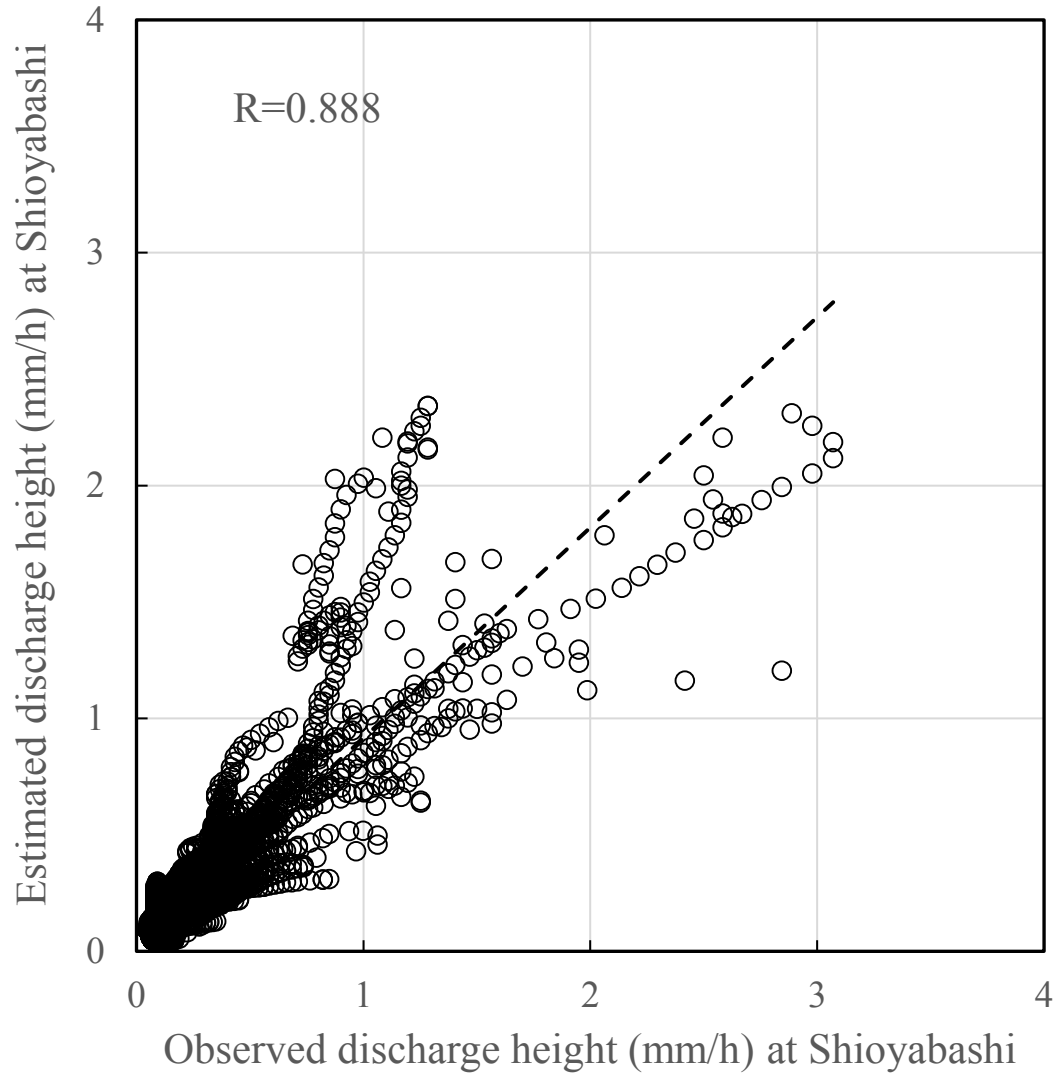

Figure 3-2. Tank model estimated with 2009 precipitation data.

3.2 Results and Discussion

3.2.1 Evaluation Metrics

NSE, RMSE, and MAE were tested on six random trainings. These random trainings were numbered from 1 to 6 and their results were shown in Figure 3-3. The results were compared with each other and indicated a basically stable relationship between the three kinds of metrics. The NSE has a negative correlation with RMSE. The trend of MAE was expected to be the same as RMSE. However, it is different between training No.2 and

training No. 3. Even if training No. 2 has higher RMSE and lower NSE than training No. 3, its MAE was lower. The hydrographs of training No. 2 and training No.3 are shown in Figure 3-4 and Figure 3-5, respectively. The blue line is observed data and the orange line is estimated data. The line of estimated data overlaps the line of observed data more in training No. 2, which has led to a result of lower MAE value than training No. 3. On the other hand, training No. 3 was able to estimate peak discharge more accurate than training No. 2, which made the RMSE of No.3 lower than training No. 2. However, these results were compared under a condition that NSE is under 0.4, which cannot be regarded as acceptable model accuracy. Since NSE can reflect the trend of RMSE and MAE, and was frequently used in recent studies, it was chosen as the evaluation metrics of this study.

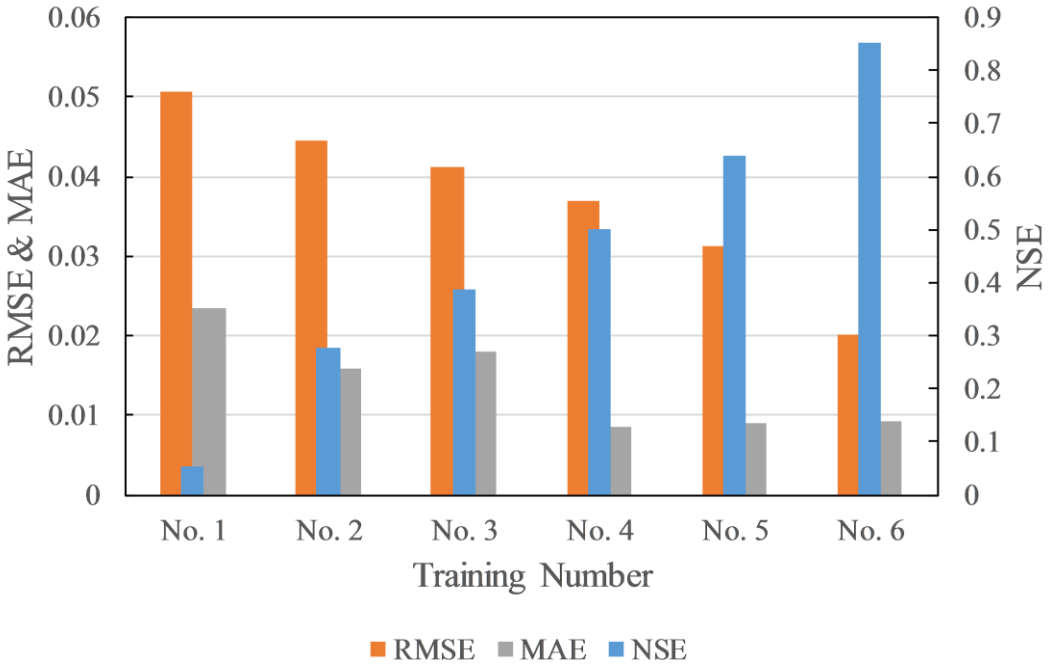

Figure 3-3. Relationship between RMSE, MAE and NSE.

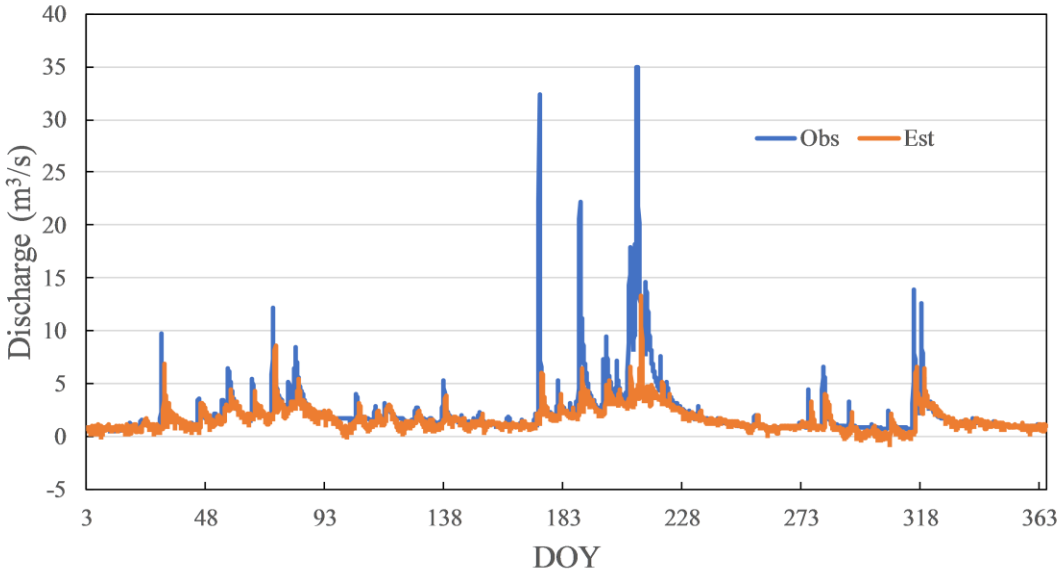

Figure 3-4. Hydrograph of Training No.2.

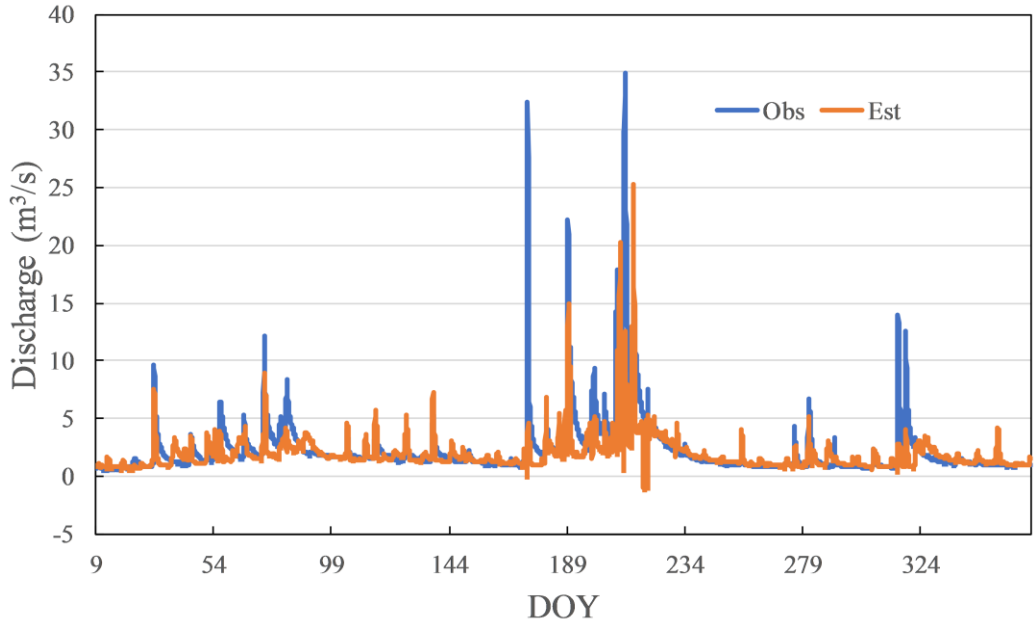

Figure 3-5. Hydrograph of Training No.3.

3.2.2 Number of Members for Ensemble Average

The P_5 of the NSE from the number of ensemble members $N = 1$ to 50 for Case 2 is shown in Figure 3-6 as an example. The blue line shows the results with $Back_{ts} = 24$, $Hid = 20$, the orange line shows the results with $Back_{ts} = 24$, $Hid = 200$, and the red line shows the results with $Back_{ts} = 168$, $Hid = 100$. At the blue line, where the number of hidden layers (Hid) is small, the accuracy is generally low, and the P_5 of NSE = 0.90 is the best. For the orange and red lines, where $Hid > 100$, increasing the number of ensemble members results in an accuracy of NSE > 0.92, which is greater than the 0.903 reference accuracy. In brief, the variation curves of P_5 of NSE become flat, and keep the accuracy level when $N \leq 20$. Thus, in this study, the training results were evaluated by the P_5 of NSE for $N = 30$ for safety to avoid the dispersion of different training results.

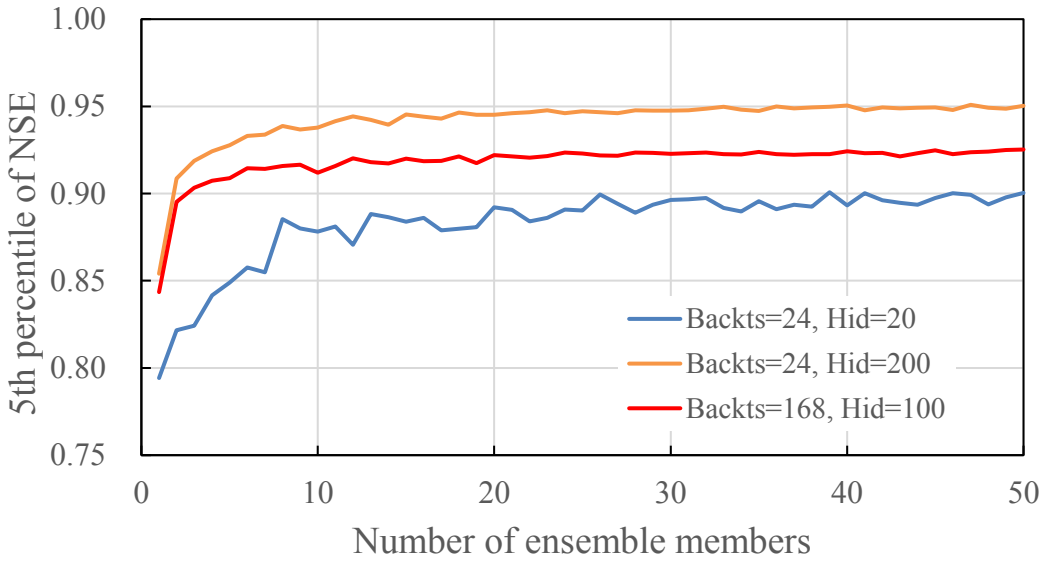

Figure 3-6. Examples of relationship between number of ensemble members (N) and 5th percentile of NSE for Case 2. Blue line: $Back_{ts}$ = 24, Hid = 20, Drp = 0, Drp_r = 0; orange line: $Back_{ts}$ = 24, Hid = 200, Drp = 0, Drp_r = 0; red line: $Back_{ts}$ = 168, Hid = 100, Drp = 0, Drp_r = 0.

3.2.3 Type of Input Variables

The P_5 of NSE, when $N = 30$, was compared to evaluate the influence of each training hyperparameter to the result. There are four

kinds of *Hid* combinations for each input variable case. Figure 3-7, Figure 3-8 and Figure 3-9 show the four combinations of the hyperparameter settings summaries of the results with $Drp = 0$ and $Drp_r = 0$. Figure 3-7 and Figure 3-8 show the results of Case 1–6 with $Back_{ts} = 24$, which assumed a 1-day missing period, and $Back_{ts} = 168$, which assumed a 7-day missing period, respectively. In Figure 3-7, Case 2, Case 3 and Case 4, which used the discharge data of both Shioyabashi and Sanpukuji as input data, were relatively good. Case 2 and Case 3 indicate 0.939 and 0.947 in the median of P_5 of NSE, respectively. They are over the reference accuracy 0.903, which is the accuracy of the linear regression model. Case 4, which used air temperature as one of the input data factors, indicates 0.900 in the median of P_5 of NSE, which was slightly less than the reference accuracy. The lower quartiles for Case 2 and Case 3 were 0.922 and 0.917, respectively, which were better than the reference accuracy. However, the minimum for Case 2 and Case 3 were 0.896 and 0.860, respectively, which were slightly less than the reference accuracy. It must be noted that Case 3, which used precipitation as one of the input data factors, has a wide variation in accuracy depending on different hyperparameter settings. In Figure 3-8, Case 2, Case 3, and Case 4 have relatively good accuracy, as is seen in Figure 3-7. Case 2 indicates 0.922 in the median of P_5 of NSE, which is over the reference accuracy. However, the median of Case 3 and Case 4 are 0.899 and 0.871, respectively, which are less than the reference accuracy. The minimum for Case 2 is 0.904, which is over the reference accuracy. Thus, Case 2 is appropriate for the input data when $Back_{ts} = 168$.

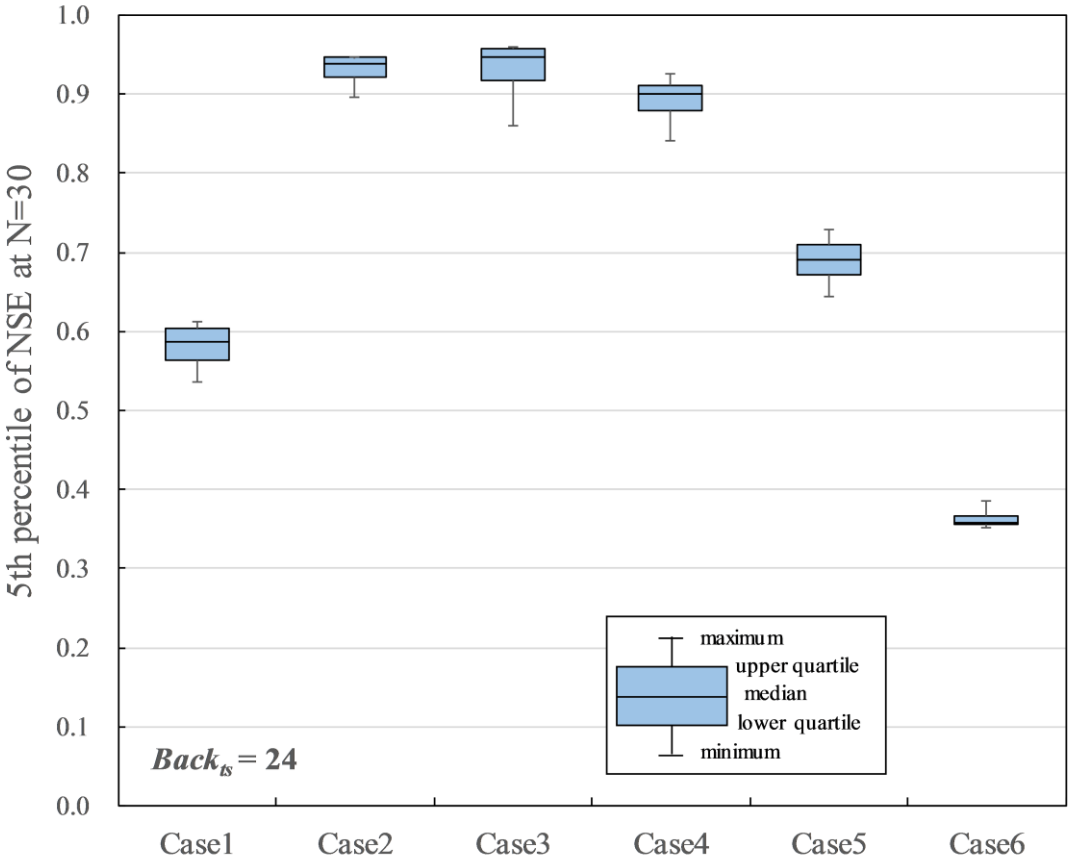

Figure 3-7. Summarized training results for Case 1 to Case 6 when $Back_{ts}$ = 24, Drp = 0, Drp_r = 0.

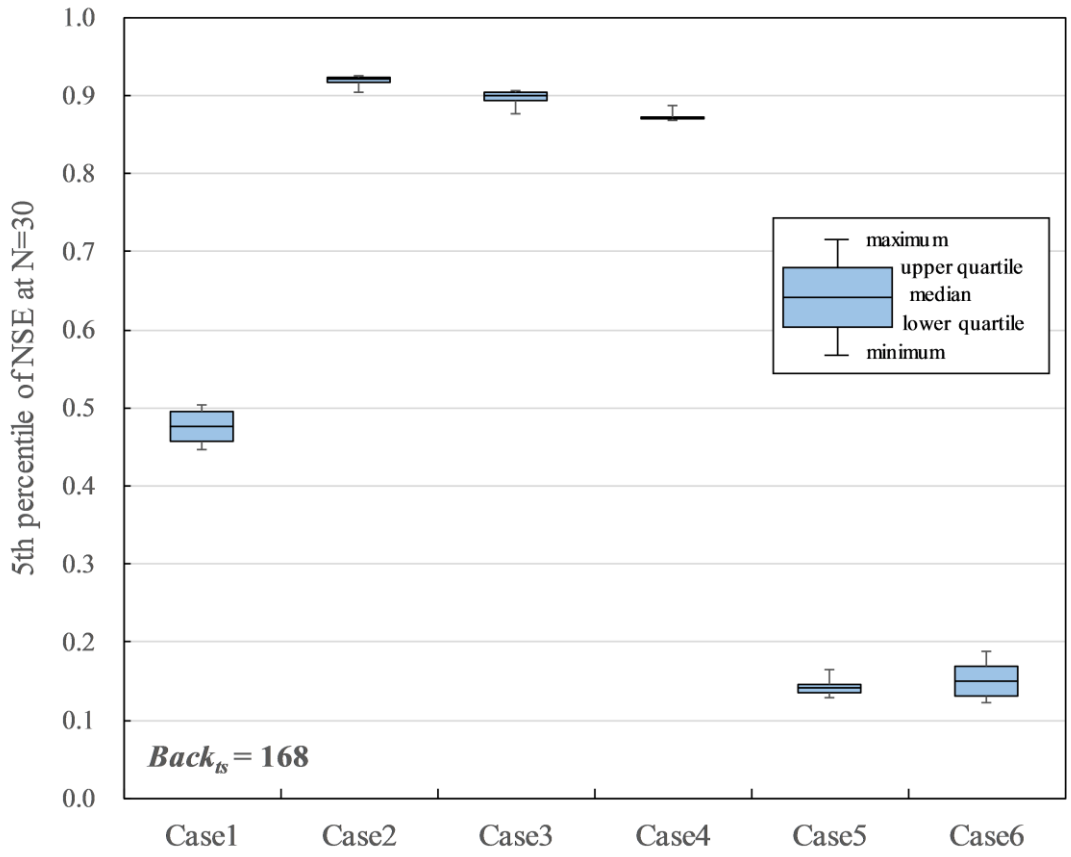

Figure 3-8. Summarized training results for Case 1 to Case 6 when $Back_{ts}$ = 168, Drp = 0, Drp_r = 0.

The results of the cases without the discharge data of Shioyabasi as $Back_{ts}$ = 0 are shown in Figure 3-9. Case 9, Case 10, and Case 11, which used discharge data of Sanpukuji as the input data, indicated relatively good accuracies. However, they were 0.877–0.891 in the median of P_5 of NSE, which was slightly less than the reference accuracy. In Case 7, where only precipitation was used as the input data, the median of the P_5 of NSE is 0.344. In Case 8, where only air temperature was used as the input data, the median of the P_5 of NSE is 0.013, indicating that it is difficult to estimate the missing data. These results suggest the understanding for the input data combination is: (i) both the Sanpukuji data and the Shioyabashi data are necessary, (ii) air temperature is not required; and (iii) precipitation may contribute to the improvement of accuracy, but it should be noted that it may cause poor accuracy depending on the parameter settings.

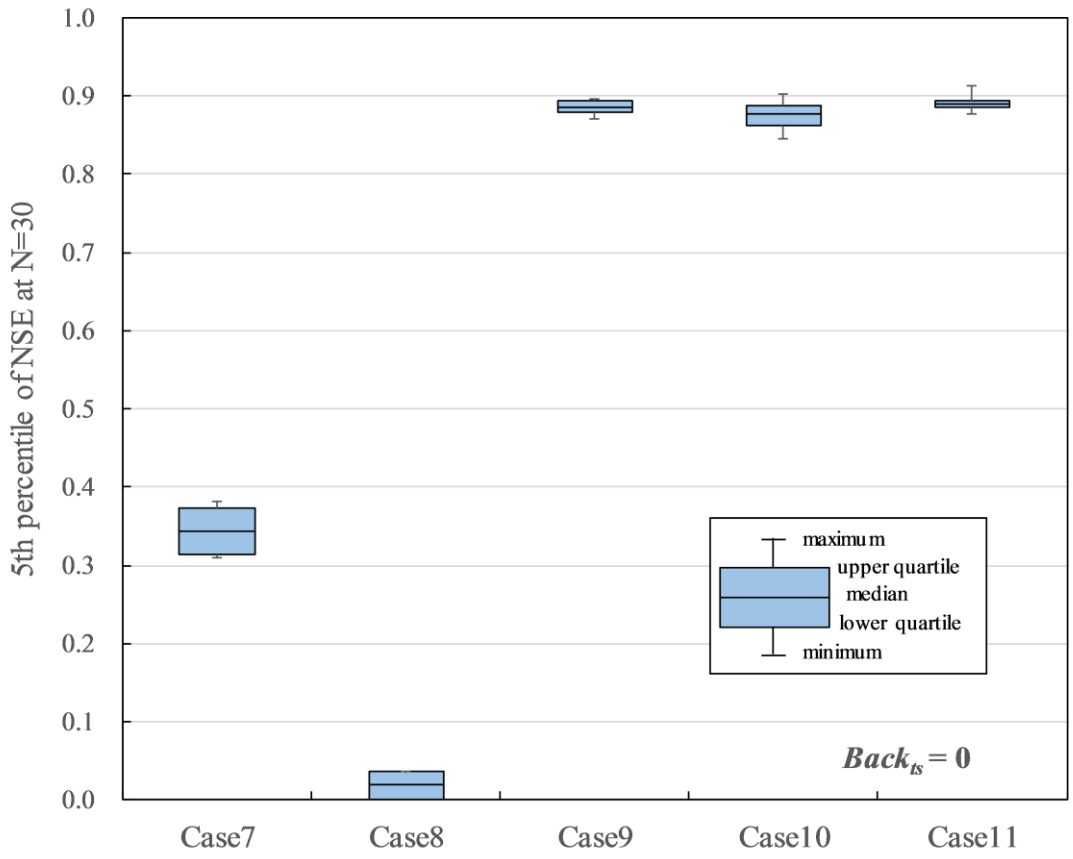

Figure 3-9. Summarized training results for Case 7 to Case 11 when $Back_{ts}$ = 0, Drp = 0, Drp_r = 0.

Two trainings of Case 1 ($Back_{ts}$ = 24, Hid = 20, Drp = 0, Drp_r = 0) and Case 8 ($Back_{ts}$ = 24, Hid = 20, Drp = 0, Drp_r = 0) were chosen to investigate the impact of precipitation to the estimation results. The hydrograph of observed discharge, precipitation, and both estimation results are shown in Figure 3-10. The blue line is observed data, and the grey line is precipitation. The green and orange lines represent discharge estimated from data with precipitation and without precipitation, respectively. The hydrograph indicates that both estimation results are responsive to precipitation events. When precipitation data was used for training, there is a trend of an occurrence of trough in the estimated discharge after a peak caused by precipitation. Additionally, the model could not estimate the discharge well when a relatively heavy precipitation event occurs. Besides, even if the location of the observation point of precipitation data is close to Sanpukuji, it is still outside of the Daihachiga

River Basin. It leads to a possibility that the precipitation data could not represent the precipitation of the whole basin of the Daihachiga River. A regional rainfall in small area near Takayama may not have influence on the discharge of upper stream. These are considered as part of the reasons that precipitation may cause lower accuracy.

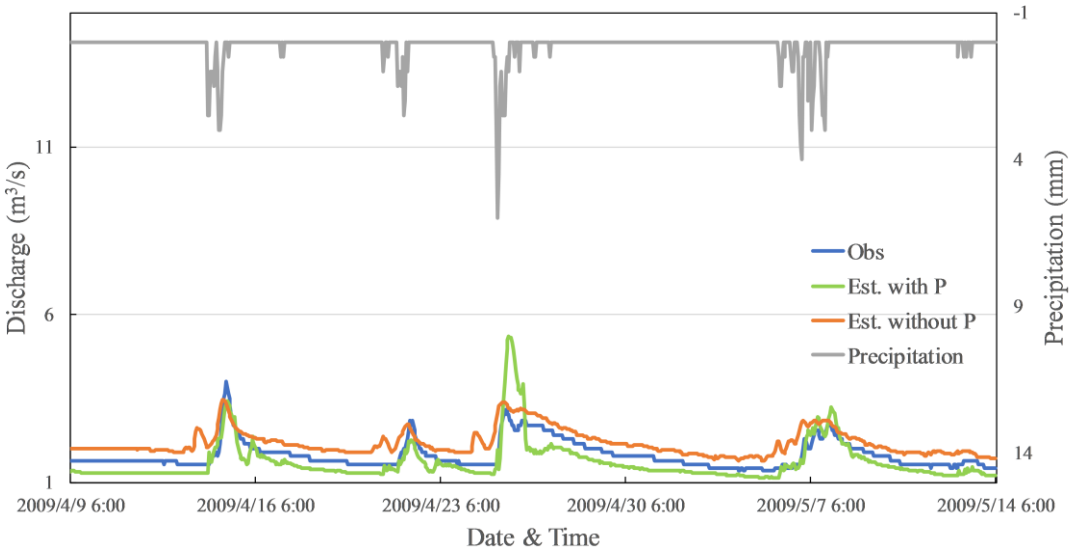

Figure 3-10. Influence of precipitation on the estimation results.

3.2.4 Dropout and Recurrent Dropout

Figure 3-11 shows the influence of dropout and recurrent dropout ($Drp\&Drp_r$), when Hid = 20 to 200 and $Back_{ts}$ = 24 for Case 3. The accuracy improved as $Drp\&Drp_r$ became smaller, and $Drp\&Drp_r$ = 0.00 had the best accuracy of 0.947 in the median P_5 of the NSE. The maximum P_5 of the NSE was 0.961 when $Drp\&Drp_r$ = 0.01 and Hid = 200. When Hid = 20, the minimums were indicated as 0.860, 0.860, 0.825, and 0.771 for $Drp\&Drp_r$ = 0.00, 0.01, 0.05, and 0.10, respectively. When $Drp\&Drp_r$ = 0.00 or 0.01, in the case of Hid > 50, the accuracies were indicated to be more than 0.920, which is over the reference accuracy. In brief, $Drp\&Drp_r$ = 0.00 shows the best results. The reason may be that higher Drp and Drp_r values drop more units for the linear transformation of the input and recurrent state. Fewer units caused a shortage of information necessary for the training, since the LSTM model of this study has only a few units in the

input and hidden layers. In the case of some other studies, a large number of units in the input and hidden layers caused drops in the units by dropout and recurrent dropout, which improved the training results [52,53]. Thus, if *Hid* is less than 200, $Drp\&Drp_r = 0.00$ is appropriate, and if *Hid* is more than 200, $Drp\&Drp_r = 0.01$ is appropriate.

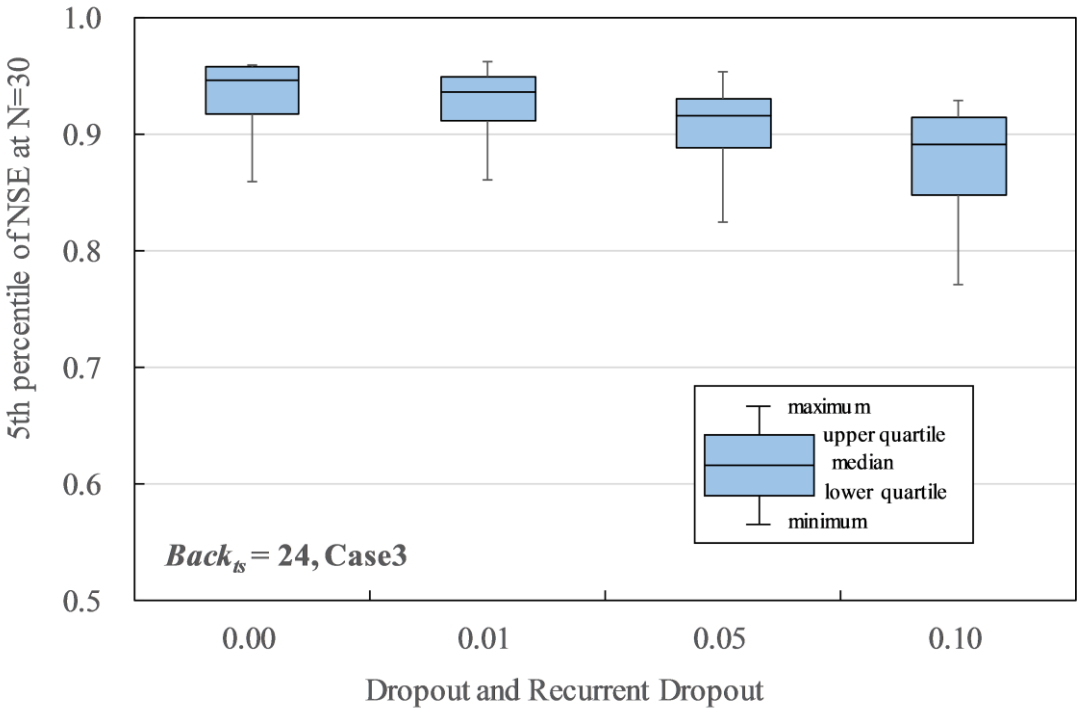

Figure 3-11. Influence of dropout and recurrent dropout on the 5th percentile of NSE.

3.2.5 Number of Hidden Layers

Figure 3-12 shows the relationship between the number of hidden layers (*Hid* = 20 to 200) and the P_5 of NSE, when $Back_{ts}$ = 24 and $Drp\&Drp_r$ = 0.00. In Cases 2, 3 and 4, the accuracy improved as *Hid* increased. Case 2 and 3 indicated better accuracy than the reference accuracy 0.903 when *Hid* > 50. However, when *Hid* > 100, the accuracies were almost the same, i.e., 0.947–0.947 for Case 2, 0.957–0.959 for Case 3. Additionally, Figure 3-13 shows the relationship between the number of hidden layers (*Hid* = 20 to 200) and the P_5 of the NSE when $Back_{ts}$ = 168 and $Drp\&Drp_r$ = 0.00. In Case 2, the accuracy slightly improved from 0.904 to 0.928 as *Hid* increased. In Case 4, the accuracy had almost no

change as *Hid* increased. On the other hand, in Case 3, *Hid* = 50 indicated the best accuracy of 0.915. However, the accuracy decreased as *Hid* increased when *Hid* > 50. In Case 3, with precipitation as the input, the estimated hydrograph might be jagged due to the influence of precipitation, showing pulsed time-series data. This is the reason why Case 3 does not always show the best accuracy. As a result, for $Back_{ts}$ = 24, *Hid* = 100 is appropriate for both Cases 2 and 3 because the accuracies for *Hid* = 100 and 200 were almost same. Setting a higher *Hid* value will just lead to a longer training time consumption. Moreover, for $Back_{ts}$ = 168, *Hid* = 100 is also appropriate for Case 2, and *Hid* = 50 is appropriate for Case 3. However, in Case 3, where precipitation is used as the input data, care should be taken because the accuracy may decrease depending on the settings.

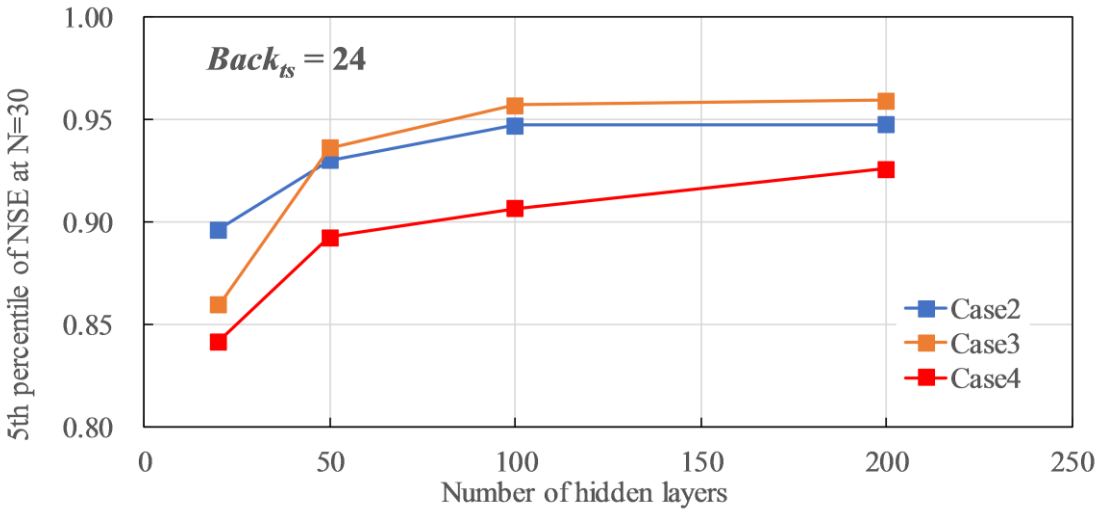

Figure 3-12. Influence of number of hidden layers to the 5th percentile of NSE, when $Back_{ts}$ = 24.

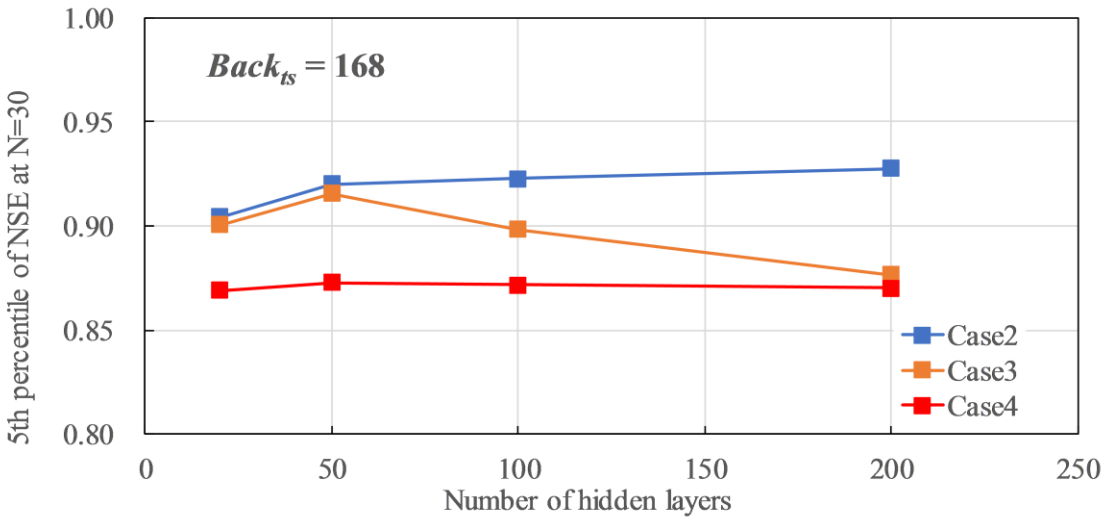

Figure 3-13. Influence of number of hidden layers to the 5th percentile of NSE, when $Back_{ts}$ = 168.

3.3 Conclusion

In this study, LSTM was applied for the imputation of missing discharge data of the Daihachiga River. The performance of an LSTM model was evaluated, and the hyperparameters of the model were tuned to satisfy the reference accuracy of 0.903 in NSE. Different hyperparameters affect the model performance to different extents. In the case of the Daihachiga River, the discharge data of both observation points should be included in input variables for training (Case 2). It is thought that the discharge data of Shioyabashi dating back 1 day or 7 days is effective in correcting the absolute value of the estimated discharge, because the estimated hydrograph with Case 9 had a slight error in base flow volume. The influence of precipitation varies. Since precipitation is strongly related to discharge, it may be useful to complete missing data. However, there is a possibility of making the estimated hydrograph jagged due to the influence of precipitation, showing pulsed time-series data. The location of observation point of precipitation data also needs to be considered. Although it is possible that filtering the precipitation data before inputting it into the LSTM will improve the accuracy, it is safer to exclude precipitation data in order to obtain consistently good results. Moreover, air temperature data could not improve the performance. Due to the small

number of units for transformation in the model of this study, setting the dropout value to 0 was suitable. A higher *Hid* value might be more suitable for the model. However, an excessive *Hid* value would just lead to a longer training time.

Consequently, this study can propose the hyperparameter settings as $Back_{ts}$ = 24, *In* = Case 2, *Hid* = 100, $Drp\&Drp_r$ = 0, a setting which is possible to estimate greater than the reference accuracy. The necessity of hyperparameter tuning was proved and the hyperparameter settings could be a reference for further research in relevant areas. Of course, this combination can be appropriately adjusted under specific experimental conditions. In future experiments, the amount of analysis data can be increased, such as the discharge data of more than two observation points, and the influence of precipitation and air temperature on the model performance can be further analyzed to improve the accuracy of the results.

Chapter 4

Conclusion

In this thesis LSTM model was applied for the imputation of missing discharge data of the Daihachiga River. The performance of the model was evaluated and compared with traditional method. There are both advantages and disadvantages for applying LSTM to such tasks. The method of improving the disadvantages was also investigated. The conclusions of this thesis are summarized as following.

Chapter 2:

- Because of the properties of SGD, the training results are dispersed. Ensemble average of the results is necessary to avoid such dispersion.
- Although there is possibility that the result of some certain training of LSTM is worse than linear regression analysis, the average of more than 20 times of training result can increase to a higher level than the linear regression analysis.
- LSTM is a feasible way to complement missing discharge data since it can obtain better accuracy than linear regression analysis even if its hyperparameters are not tuned. The model needs more improvement to make it applicable.
- Using the data of neighboring observation point is useful for improving the estimation accuracy.

Chapter 3:

- Hyperparameters can affect the LSTM model performance in different extent.
- Using discharge data of both observation points is the best choice in the case of the Daihachiga River.
- Precipitation data could be noise. In some cases, it can improve the performance. But in other cases, it may lower the estimation accuracy. It might be caused by the estimation mechanism of the LSTM model. Besides, the location of observation point of precipitation might lead to

a result that in certain rainfall events the precipitation has no influence on the discharge of upper stream.

- Air temperature almost has no influence on the results
- High dropout values may cause information shortage at transformation of cell state to a model with few units. It leads to decrease of the model performance.
- The LSTM model can complement the 24 hours of missing data gap very well. If the gap is 168 hours long, the accuracy is acceptable. But at the same time, setting of dropout needs special attention.
- Although more hidden layer leads to better results, time consumption of training needs to be considered when setting the number of hidden layers.
- Optimal hyperparameter setting can improve the model performance significantly. In this study, the optimal hyperparameter setting is $Back_{ts}$ = 24, *In* = Case 2, *Hid* = 100, $Drp\&Drp_r$ = 0.

The LSTM has its specific inherent advantages over traditional methods. It was proved competent and had better performance than traditional method in imputation of missing discharge data. In this study, the potential of the LSTM for imputation of missing data was underlined, and feasible application method to further research of missing hydrological data imputation was provided. Those advantages of the LSTM should be explored and applied on not only missing discharge data, but also other kinds of hydrological data such as water level. In future studies, the application method of the LSTM should also be improved to adjust the new application scenarios.

Bibliography

1. Ministry of Land Infrastructure Transport and Tourism 川の防災情報[Disaster Information for River] Available online: https://www.river.go.jp/index (accessed on 20 October 2021).
2. Kodaka, S. 近畿地方整備局における河川情報の監視と水文観測データの品質照査について[Monitoring of River Information and Quality Verification of Hydrological Observation Data in Kinki Regional Development Bureau]. In Proceedings of the 平成25年度近畿地方整備局研究発表会論文集 [Proceedings of the Annual Conference of Ministry of Land, Infrastructure, Transport and Tourism Kinki Regional Development Bureau]; Osaka, July 12 2013.
3. Yamaguchi, N.; Shimodori, T.; Higuchi, M.; Honte, R. 明神雨量テレメータの欠測改善検討について[Study on the Improvement of Missing Measurements of the Myojin Rainfall Telemeter]. In Proceedings of the 平成29年度北陸地方整備局事業研究発表会[Proceedings of the annual conference of Ministry of Land, Infrastructure, Transport and Tourism Hokuriku Regional Development Bureau]; Niigata, July 25 2017.
4. Aburatani, R.; Ohgushi, K.; Tezuka, M. 有明海流入河川の流量欠測データの補間方法について[Interpolation Method for Missing Flow Data of Rivers Flowing into the Ariake Sea]. In Proceedings of the Proceedings of Japan Society of Civil Engineers-west Annual Meeting; Kagoshima, 2011; pp. 155–156.
5. Kojiri, T.; Panu, U.S.; Tomosugi, K. Complement Method of Observation Lack of Discharge with Pattern Classification and Fuzzy Inference. *Journal of Japan Society of Hydrology and Water Resources* **1994**, *7*, 536–543, doi:10.3178/jjshwr.7.6_536.

6. Inayoshi, A.; Nagae, K.; Tamiya, M.; Miyata, T.; Shu-ichi, M.; Takemura, H. Basic Study of the Use of a Neural Network Model to Predict Flooding on Second Class Rivers. *Advances in River Engineering* **2003**, *9*, 179–184.

7. Abe, K.; Kikuchi, H.; Furukawa, K.; Shiotsuki, Y. STUDY ON A SYSTEM OF RUNOFF (DAILY) ANALYSIS USING NEURAL NETWORK. *Doboku Gakkai Ronbunshu* **2000**, *2000*, 1–13, doi:10.2208/jscej.2000.656_1.

8. Wada, K.; Nohara, D.; Kojiri, T. 気象情報と人工知能手法を利用した長期流量予測に関する研究[Research on Long-Term Flow Forecasting Using Meteorological Information and Artificial Intelligence Methods]. *Proceedings of annual conference, Japan Society of Hydrology and Water Resources* **2006**, *19*, 66–67, doi:10.11520/jshwr.19.0.33.0.

9. Hitokoto, M.; Sakuraba, M. HYBRID DEEP NEURAL NETWORK AND DISTRIBUTED RAINFALL-RUNOFF MODEL FOR THE REAL-TIME RIVER STAGE PREDICTION. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)* **2017**, *73*, 22–33, doi:10.2208/jscejhe.73.22.

10. Hitokoto, M.; Sakuraba, M.; Sei, Y. Development of the Real-Time River Stage Prediction Method Using Deep Learning. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)* **2016**, *72*, I_187-I_192, doi:10.2208/jscejhe.72.I_187.

11. Fusamae, K.; Shimamoto, T. AIを活用した洪水予測技術の開発について[Development of AI-Based Flood Forecasting Technology]. In Proceedings of the Proceedings of Research Meeting, Ministry of Land, Infrastructure, Transport and Tourism Kyushu Regional Development Bureau; Hakata, 2019.

12. Kimura, N.; Nakada, T.; Azechi, I.; Sekijima, K.; Kiri, H.; Baba, D. Prediction on Water Levels in a Wet Pond for a Drainage System Using an Artificial Neural Network Model. *Bulletin of the NARO, Rural Engineering* **2019**, *3*, 71–80, doi:10.24514/00001157.

13. Tamura, K.; Kanou, S.; Miura, S.; Yamawaki, M.; Kaneko, H. Application of Deep Learning to Long-Term Prediction of Dam Inflow Toward Efficiency of the Flood Control Operation. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)* **2018**, *74*, I_1327-I_1332, doi:10.2208/jscejhe.74.5_I_1327.

14. 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 (Switzerland)* **2020**, *12*, 96, doi:10.3390/w12010096.

15. Teranishi, T.; Shidawara, M. A Consideration on the Recurrent Structure of Neural Networks for Runoff Analysis. *Bulletin of Aichi Institute of Technology* **1997**, *32*, 89–96.

16. Taniguchi, J.; Kojima, T.; Sota, Y.; Hukumoto, S.; Satou, H.; Machida, Y.; Mikami, T.; Nagayama, M.; Nishikohri, T.; Wwatanabe, A. Application of Recurrent Neural Network for Dam Inflow Prediction. *Advances in River Engineering* **2019**, *25*.

17. Chiang, Y.M.; Chang, L.C.; Tsai, M.J.; Wang, Y.F.; Chang, F.J. Dynamic Neural Networks for Real-Time Water Level Predictions of Sewerage Systems-Covering Gauged and Ungauged Sites. *Hydrology and Earth System Sciences* **2010**, *14*, 1309–1319, doi:10.5194/hess-14-1309-2010.

18. Sahoo, A.; Samantaray, S.; Ghose, D.K. Stream Flow Forecasting in Mahanadi River Basin Using Artificial Neural Networks. *Procedia Computer Science* **2019**, *157*, 168–174, doi:10.1016/j.procs.2019.08.154.

19. Sit, M.; Demiray, B.Z.; Xiang, Z.; Ewing, G.J.; Sermet, Y.; Demir, I. A Comprehensive Review of Deep Learning Applications in Hydrology and Water Resources. *Water Science and Technology* **2020**, *82*, 2635–2670, doi:10.2166/wst.2020.369.

20. Hu, C.; Wu, Q.; Li, H.; Jian, S.; Li, N.; Lou, Z. Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *Water (Switzerland)* **2018**, *10*, doi:10.3390/w10111543.

21. Lee, G.; Lee, D.; Jung, Y.; Kim, T.-W. Comparison of Physics-Based and Data-Driven Models for Streamflow Simulation of the Mekong River. *Journal of Korea Water Resources Association* **2018**, *51*, 503–514, doi:10.3741/JKWRA.2018.51.6.503.

22. Lee, T.; Shin, J.Y.; Kim, J.S.; Singh, V.P. Stochastic Simulation on Reproducing Long-Term Memory of Hydroclimatological Variables Using Deep Learning Model. *Journal of Hydrology* **2020**, *582*, doi:10.1016/j.jhydrol.2019.124540.

23. Xiang, Z.; Yan, J.; Demir, I. A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning. *Water Resources Research* **2020**, *56*, doi:10.1029/2019WR025326.

24. Fan, H.; Jiang, M.; Xu, L.; Zhu, H.; Cheng, J.; Jiang, J. Comparison of Long Short Term Memory Networks and the Hydrological Model in Runoff Simulation. *Water (Switzerland)* **2020**, *12*, doi:10.3390/w12010175.

25. Bai, P.; Liu, X.; Xie, J. Simulating Runoff under Changing Climatic Conditions: A Comparison of the Long Short-Term Memory Network with Two Conceptual Hydrologic Models. *Journal of Hydrology* **2021**, *592*, doi:10.1016/j.jhydrol.2020.125779.

26. Kratzert, F.; Klotz, D.; Brenner, C.; Schulz, K.; Herrnegger, M. Rainfall-Runoff Modelling Using Long Short-Term Memory (LSTM) Networks. *Hydrology and Earth System Sciences* **2018**, *22*, 6005–6022, doi:10.5194/hess-22-6005-2018.

27. Sudriani, Y.; Ridwansyah, I.; A Rustini, H. Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) for Discharge Level Prediction and Forecast in Cimandiri River, Indonesia. In Proceedings of the IOP Conference Series: Earth and Environmental Science; Institute of Physics Publishing, July 29 2019; Vol. 299.

28. Ding, Y.; Zhu, Y.; Feng, J.; Zhang, P.; Cheng, Z. Interpretable Spatio-Temporal Attention LSTM Model for Flood Forecasting. *Neurocomputing* **2020**, *403*, 348–359, doi:10.1016/j.neucom.2020.04.110.

29. Le, X.H.; Ho, H.V.; Lee, G.; Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water (Switzerland)* **2019**, *11*, doi:10.3390/w11071387.

30. Li, W.; Kiaghadi, A.; Dawson, C. Exploring the Best Sequence LSTM Modeling Architecture for Flood Prediction. *Neural Computing and Applications* **2020**, doi:10.1007/s00521-020-05334-3.

31. Zhang, J.; Zhu, Y.; Zhang, X.; Ye, M.; Yang, J. Developing a Long Short-Term Memory (LSTM) Based Model for Predicting Water Table Depth in Agricultural Areas. *Journal of Hydrology* **2018**, *561*, 918–929, doi:10.1016/j.jhydrol.2018.04.065.

32. Sahoo, B.B.; Jha, R.; Singh, A.; Kumar, D. Long Short-Term Memory (LSTM) Recurrent Neural Network for Low-Flow Hydrological Time Series Forecasting. *Acta Geophysica* **2019**, *67*, 1471–1481, doi:10.1007/s11600-019-00330-1.

33. Qin, J.; Liang, J.; Chen, T.; Lei, X.; Kang, A. Simulating and Predicting of Hydrological Time Series Based on TensorFlow Deep Learning. *Polish Journal of Environmental Studies* **2019**, *28*, 795–802, doi:10.15244/pjoes/81557.

34. Granata, F.; di Nunno, F. Forecasting Evapotranspiration in Different Climates Using Ensembles of Recurrent Neural Networks. *Agricultural Water Management* **2021**, *255*, 107040, doi:10.1016/j.agwat.2021.107040.

35. Chen, Z.; Zhu, Z.; Jiang, H.; Sun, S. Estimating Daily Reference Evapotranspiration Based on Limited Meteorological Data Using Deep Learning and Classical Machine Learning Methods. *Journal of Hydrology* **2020**, *591*, 125286, doi:10.1016/j.jhydrol.2020.125286.

36. Ferreira, L.B.; da Cunha, F.F. Multi-Step Ahead Forecasting of Daily Reference Evapotranspiration Using Deep Learning. *Computers and Electronics in Agriculture* **2020**, *178*, 105728, doi:10.1016/j.compag.2020.105728.

37. Reimers, N.; Gurevych, I. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. **2017**.

38. Hossain, M.D.; Ochiai, H.; Fall, D.; Kadobayashi, Y. LSTM-Based Network Attack Detection: Performance Comparison by Hyper-Parameter Values Tuning. In Proceedings of the 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom); 2020; pp. 62–69.

39. Yadav, A.; Jha, C.K.; Sharan, A. Optimizing LSTM for Time Series Prediction in Indian Stock Market. *Procedia Computer Science* **2020**, *167*, 2091–2100, doi:10.1016/j.procs.2020.03.257.

40. Yi, H.; Bui, K.N. An Automated Hyperparameter Search-Based Deep Learning Model for Highway Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems* **2020**, 1–10, doi:10.1109/TITS.2020.2987614.

41. Kratzert, F.; Klotz, D.; Shalev, G.; Klambauer, G.; Hochreiter, S.; Nearing, G. Benchmarking a Catchment-Aware Long Short-Term Memory Network (LSTM) for Large-Scale Hydrological Modeling. *Hydrology and Earth System Sciences Discussions* **2019**, 1–32, doi:10.5194/hess-2019-368.

42. Afzaal, H.; Farooque, A.A.; Abbas, F.; Acharya, B.; Esau, T. Groundwater Estimation from Major Physical Hydrology Components Using Artificial Neural Networks and Deep Learning. *Water (Switzerland)* **2020**, *12*, doi:10.3390/w12010005.

43. Ayzel, G.; Heistermann, M. The Effect of Calibration Data Length on the Performance of a Conceptual Hydrological Model versus LSTM and GRU: A Case Study for Six Basins from the CAMELS Dataset. *Computers and Geosciences* **2021**, *149*, 104708, doi:10.1016/j.cageo.2021.104708.

44. Alizadeh, B.; Ghaderi Bafti, A.; Kamangir, H.; Zhang, Y.; Wright, D.B.; Franz, K.J. A Novel Attention-Based LSTM Cell Post-Processor Coupled with Bayesian Optimization for Streamow Prediction. *Journal of Hydrology* **2021**, *601*, 126526, doi:10.1016/j.jhydrol.2021.126526.

45. Dastorani, M.T.; Moghadamnia, A.; Piri, J.; Rico-Ramirez, M. Application of ANN and ANFIS Models for Reconstructing Missing Flow Data. *Environmental Monitoring and Assessment* **2010**, *166*, 421–434, doi:10.1007/s10661-009-1012-8.

46. Mispan, M.R.; Rahman, N.F.A.; Ali, M.F.; Khalid, K.; Bakar, M.H.A.; Haron, S.H. Missing River Discharge Data Imputation Approach Using Artificial Neural Network. *ARPN Journal of Engineering and Applied Sciences* **2015**, *10*, 10480–10485.

47. Gers, F.A.; Schmidhuber, J.A.; Cummins, F.A. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* **2000**, *12*, 2451–2471, doi:10.1162/089976600300015015.

48. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780, doi:10.1162/neco.1997.9.8.1735.

49. River Division of Gifu Prefectural Office 大島ダム[Ojima Dam] Available online: https://www.pref.gifu.lg.jp/page/67841.html.

50. Kojima, T.; Shinoda, S.; Mahboob, M.G.; Ohashi, K. Study on Improvement of Real-Time Flood Forecasting with Rainfall Interception Model. *Advances in River Engineering* **2012**, *18*, 435–440.

51. Nash, J.E.; Sutcliffe, J. v River Flow Forecasting through Conceptual Models Part I — A Discussion of Principles. *Journal of Hydrology* **1970**, *10*, 282–290, doi:https://doi.org/10.1016/0022-1694(70)90255-6.

52. Semeniuta, S.; Severyn, A.; Barth, E. Recurrent Dropout without Memory Loss. In Proceedings of the Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers; Matsumoto, Y., Prasad, R., Eds.; The COLING 2016 Organizing Committee: Osaka, December 2016; pp. 1757–1766.

53. Gal, Y.; Ghahramani, Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In Proceedings of the Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16); Curran Associates Inc.: Barcelona, December 2016; pp. 1027–1035.

Appendix

Table A1. Table of 90 kinds of hyperparameter combination

Training No.	*In*	$Back_{ts}$	*Hid*	*Drp*	Drp_r	P_5 of NSE
1	$Q_{shio} + Q_{san}$	24	20	0	0	0.896323
2	$Q_{shio} + P$	24	20	0	0	0.612851
3	$Q_{shio} + Q_{san} + P$	24	20	0	0	0.859700
4	$Q_{shio} + Q_{san} + P + T$	24	20	0	0	0.841739
5	Q_{shio}	24	20	0	0	0.643958
6	$Q_{shio} + T$	24	20	0	0	0.351267
7	$Q_{shio} + Q_{san} + P$	24	20	0.01	0.01	0.860271
8	$Q_{shio} + Q_{san} + P$	24	20	0.05	0.05	0.824750
9	$Q_{shio} + Q_{san} + P$	24	20	0.1	0.1	0.771259
10	$Q_{shio} + Q_{san}$	24	50	0	0	0.930278
11	$Q_{shio} + P$	24	50	0	0	0.573846
12	$Q_{shio} + Q_{san} + P$	24	50	0	0	0.936166
13	$Q_{shio} + Q_{san} + P + T$	24	50	0	0	0.892641
14	Q_{shio}	24	50	0	0	0.680160
15	$Q_{shio} + T$	24	50	0	0	0.356994
16	$Q_{shio} + Q_{san} + P$	24	50	0.01	0.01	0.927786
17	$Q_{shio} + Q_{san} + P$	24	50	0.05	0.05	0.908147
18	$Q_{shio} + Q_{san} + P$	24	50	0.1	0.1	0.873763
19	$Q_{shio} + Q_{san}$	24	100	0	0	0.947078
20	$Q_{shio} + P$	24	100	0	0	0.535739
21	$Q_{shio} + Q_{san} + P$	24	100	0	0	0.957068
22	$Q_{shio} + Q_{san} + P + T$	24	100	0	0	0.906750
23	Q_{shio}	24	100	0	0	0.727970
24	$Q_{shio} + T$	24	100	0	0	0.359394
25	$Q_{shio} + Q_{san} + P$	24	100	0.01	0.01	0.943825
26	$Q_{shio} + Q_{san} + P$	24	100	0.05	0.05	0.922461
27	$Q_{shio} + Q_{san} + P$	24	100	0.1	0.1	0.909169
28	$Q_{shio} + Q_{san}$	24	200	0	0	0.947493
29	$Q_{shio} + P$	24	200	0	0	0.601911
30	$Q_{shio} + Q_{san} + P$	24	200	0	0	0.959165
31	$Q_{shio} + Q_{san} + P + T$	24	200	0	0	0.925806
32	Q_{shio}	24	200	0	0	0.702479
33	$Q_{shio} + T$	24	200	0	0	0.385552
34	$Q_{shio} + Q_{san} + P$	24	200	0.01	0.01	0.961166
35	$Q_{shio} + Q_{san} + P$	24	200	0.05	0.05	0.953193
36	$Q_{shio} + Q_{san} + P$	24	200	0.1	0.1	0.928058
37	$Q_{shio} + Q_{san}$	168	20	0	0	0.904154
38	$Q_{shio} + P$	168	20	0	0	0.499526
39	$Q_{shio} + Q_{san} + P$	168	20	0	0	0.900507
40	$Q_{shio} + Q_{san} + P + T$	168	20	0	0	0.869109
41	Q_{shio}	168	20	0	0	0.129549

42	$Q_{shio} + T$	168	20	0	0	0.122250
43	$Q_{shio} + Q_{san}$	168	50	0	0	0.920067
44	$Q_{shio} + P$	168	50	0	0	0.494320
45	$Q_{shio} + Q_{san} + P$	168	50	0	0	0.915458
46	$Q_{shio} + Q_{san} + P + T$	168	50	0	0	0.872925
47	Q_{shio}	168	50	0	0	0.138257
48	$Q_{shio} + T$	168	50	0	0	0.133258
49	$Q_{shio} + Q_{san}$	168	50	0.01	0.01	0.918030
50	$Q_{shio} + Q_{san} + P$	168	50	0.01	0.01	0.914554
51	$Q_{shio} + Q_{san}$	168	50	0.05	0.05	0.900544
52	$Q_{shio} + Q_{san}$	168	50	0.1	0.1	0.874470
53	$Q_{shio} + Q_{san}$	168	100	0	0	0.922857
54	$Q_{shio} + P$	168	100	0	0	0.446777
55	$Q_{shio} + Q_{san} + P$	168	100	0	0	0.898342
56	$Q_{shio} + Q_{san} + P + T$	168	100	0	0	0.871731
57	Q_{shio}	168	100	0	0	0.146017
58	$Q_{shio} + T$	168	100	0	0	0.165140
59	$Q_{shio} + Q_{san} + P$	168	100	0.01	0.01	0.906672
60	$Q_{shio} + Q_{san} + P$	168	100	0.05	0.05	0.904354
61	$Q_{shio} + Q_{san} + P$	168	100	0.1	0.1	0.887407
62	$Q_{shio} + Q_{san}$	168	200	0	0	0.927516
63	$Q_{shio} + P$	168	200	0	0	0.459666
64	$Q_{shio} + Q_{san} + P$	168	200	0	0	0.876401
65	$Q_{shio} + Q_{san} + P + T$	168	200	0	0	0.869963
66	Q_{shio}	168	200	0	0	0.146245
67	$Q_{shio} + T$	168	200	0	0	0.182535
68	$Q_{shio} + Q_{san} + P$	168	200	0.01	0.01	0.883799
69	$Q_{shio} + Q_{san} + P$	168	200	0.05	0.05	0.883235
70	$Q_{shio} + Q_{san} + P$	168	200	0.1	0.1	0.881285
71	Q_{san}	None	20	0	0	0.880953
72	P	None	20	0	0	0.374783
73	T	None	20	0	0	-0.016940
74	$Q_{san} + P$	None	20	0	0	0.892470
75	$Q_{san} + T$	None	20	0	0	0.892694
76	Q_{san}	None	50	0	0	0.871054
77	P	None	50	0	0	0.315801
78	T	None	50	0	0	-0.003190
79	$Q_{san} + P$	None	50	0	0	0.885649
80	$Q_{san} + T$	None	50	0	0	0.876715
81	Q_{san}	None	100	0	0	0.888463
82	P	None	100	0	0	0.310116
83	T	None	100	0	0	0.029295
84	$Q_{san} + P$	None	100	0	0	0.867375
85	$Q_{san} + T$	None	100	0	0	0.889015
86	Q_{san}	None	200	0	0	0.909590
87	P	None	200	0	0	0.372668
88	T	None	200	0	0	0.032621

89	$Q_{san} + P$	None	200	0	0	0.845648
90	$Q_{san} + T$	None	200	0	0	0.900389