Deep-learning algorithm development for river flow prediction: PNP algorithm

Gwiman Bak
Chonnam National University
Youngchul Bae (ycbae@chonnam.ac.kr)
Chonnam National University https://orcid.org/0000-0003-3184-9667

Research Article

Keywords: Deep learning, PNP, LSTM, River flow, Prediction

Posted Date: August 8th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1916592/v1

Version of Record: A version of this preprint was published at Soft Computing on May 13th, 2023. See the published version at https://doi.org/10.1007/s00500-023-08254-1.

Abstract

Deep-learning algorithms developed in recent decades have performed well in prediction and classification using accumulated big data. However, as climate change has recently become a more serious global problem, natural disasters are occurring frequently. When analyzing natural disasters from the perspective of a data analyst, they are considered outliers, and the ability to predict outliers (natural disasters) using deep-learning algorithms based on big data acquired by computers is limited. To predict natural disasters, deep-learning algorithms must be enhanced to be able to predict outliers based on information such as the correlation between the input and output. Thus, algorithms that specialize in one field must be developed, and specialized algorithms for abnormal values must be developed to predict natural disasters. Therefore, considering the correlation between the input and output, we propose a positive and negative perceptron (PNP) algorithm to predict the flow rate of rivers using climate change-sensitive precipitation. The PNP algorithm consists of a hidden deep-learning layer composed of positive and negative neurons. We built deep-learning models using the PNP algorithm to predict the flow of three rivers. We also built comparative deep-learning models using long short-term memory (LSTM) to validate the performance of the PNP algorithm. We compared the predictive performance of each model using the root mean square error and symmetric mean absolute percentage error and demonstrated that it performed better than the LSTM algorithms.

1 Introduction

Deep-learning technology is emerging all around us, strengthening the technical aspects of modern society, from time-series prediction to image analysis to natural language processing[1–7]. This technology has been widely used recently in biological mechanisms, helping to solve problems that have previously been difficult to solve[8–12]. In addition, better algorithms have been developed for time-series prediction, such as recurrent neural networks (RNN), long short-term memory (LSTM), and gated recurrent units (GRU), which have shown strength in time-series predictions using deep learning based on big data accumulated in recent years[13–18]. To date, deep-learning algorithms have been developed by focusing on the structure of the human brain, and the algorithms thus developed have generally performed well. The learning method of deep learning varies depending on the composition of the data because the method creates an optimal model based on the data. In the process of acquiring data, abnormal data may occur, such as those resulting from errors in sensors or abnormalities in the acquisition of the data. These are called outliers. Deep-learning models are affected by learning performance if there is an outlier in the learning data. Many researchers training deep-learning models remove outliers in the data to improve the learning performance of the models[19–21]. From an analyst's point of view in the environmental field, natural disasters are judged to be outliers because the period of data acquisition using sensors is much shorter than the longest patterns of the earth. These natural disasters occurred once in 100 years in the past, but they have occurred more frequently in recent years due to climate change. If deep-learning expert judges natural-disaster data as outliers and delete them to training deep-learning model, that can’t predict natural disasters. For this reason, deep-learning algorithms for

environmental data should be developed to predict outliers (natural disasters) and provide information such as the causes of the natural disasters that the deep-learning model is trying to predict and their correlation with the input to the deep-learning model. To do this, it is necessary to identify the correlation and characteristics of the input and output of deep-learning models and apply them to deep-learning algorithms. In addition, to predict outliers (natural disasters), deep-learning algorithms specializing in a single field and specialized algorithms for outlier values must be developed. As the climate changes, natural disasters such as floods and droughts occur more frequently, and these disasters seriously impact cities around rivers. Climate change is one of the major problems facing humans because the changing climate patterns are expected to increase the frequency of extreme weather events[22, 23]. In recent years, the frequency of droughts caused by global warming has been increasing. River and groundwater levels are declining due to the increased water demand[24]. As a result, water shortages, ground subsidence, and penetration into the groundwater layer of seawater may occur[25–27]. In addition, global warming can add to the collision of cold and dry air currents at high latitudes and warm and humid air currents at low latitudes, causing frequent torrential rains and floods worldwide[28–31]. For this reason, it is necessary to analyze and predict the flow of rivers to prevent damage from water depletion and natural disasters from floods. Therefore, we present a positive and negative perceptron (PNP) algorithm that can predict the river flow using precipitation sensitive to climate change considering the correlation between the input and output.

This paper is organized as follows. Section 2 introduces the deep-learning algorithm used in time-series prediction and the study of predicting river flow through deep learning. Section 3 describes the introduction, composition, and formula of the PNP algorithm. Section 4 describes the input and flow data to be used in Sections 5 and 6. Section 5 measures the performance of the PNP algorithm, proving that it outperforms the LSTM algorithm. Section 6 predicts the river flow using not only the PNP algorithm but also the LSTM based on several variables. Finally, Section 7 comments on the predictive results and provides directions for future research.

2 Related Works

In this section, we introduce the deep-learning algorithm used in time-series prediction and the study of predicting river flow through deep learning. The artificial neural network (ANN), the most basic concept of deep learning, was inspired by biological processes in the 1960s, when it was discovered that different visual cortical cells were activated when cats visualize different objects[32, 33]. The study reviewed the links between the eyes and cells in the visual cortex and showed that information was processed hierarchically in the visual system. The ANN mimicked the recognition of objects by connecting artificial neurons inside layers that could extract object features[34]. An ANN does not provide information on the sequential order of the input, so there are limitations in processing time-series data with dependencies between the data. Time-series data or strings are generally used to predict later data by the data entered earlier. Therefore, it is difficult to accurately predict time-series data using an ANN, which only disseminates the current input data in the order of input, hidden, and output layers. To address these problems, the RNN was proposed, an ANN designed to process temporally continuous data, such as time-

series data[35]. The RNN adds loop connections to each neuron in the hidden layer to input the hidden layer output from previous data back into the hidden layer neurons when predicting data from the present time. This allows an RNN to make predictions about current data simultaneously based on the data entered at a previous time. However, when backpropagation algorithms learn data covering a long period, the gradient can be abnormally increased or reduced. LSTM was proposed to solve the gradient problems that occur with RNNs, and most RNN-based applications proposed to date have been implemented using LSTM[36]. LSTM adds new elements such as a forget gate, input gate, and output gate to each neuron in the hidden layer to solve gradient-related problems. As the layer deepens, LSTM causes information loss because the encoder has too much information to compress and tends to use the information compressed by the encoder only for the initial prediction. Thus, bidirectional RNN (BRNN) and bidirectional LSTM (BLSTM) were developed[37]. Two-way BRNN and BLSTM add a backward processing layer to the existing RNN and LSTM layers, where the input data are transmitted to both forward and backward learning. This does not degrade the performance, even if the data are lengthy. LSTM has become a model that performs well even with data with long sequences while also solving the long-term dependence problem, but it requires more parameters than an RNN due to its complex structure. Overfitting can occur when there are not enough learning data. GRUs have been proposed to improve on these shortcomings[38].

Table 1
Related deep-learning algorithms

Num	Year	Algorithm	Abbreviation	Reference
1	1958	Multi-layer perceptron	MLP	[34]
2	1990	Recurrent neural network	RNN	[35]
3	1997	Long short-term memory	LSTM	[36]
4	1997	Bidirectional recurrent neural network	BRNN	[37]
		Bidirectional long short-term memory	BLSTM	
5	2014	Gated recurrent unit	GRU	[38]
6	2022	Positive & negative perceptron	PNP	-

The following is a list of flow prediction studies using deep learning. Fathian et al. analyzed time series using self-exciting threshold autoregressive (SETAR) and generalized autoregressive conditional heteroscedasticity (GARCH) models. Then they used multivariable adaptive regression splines (MARS) and random forest (RF) models to predict the monthly river flow of the Grand River's Brantford and Galt Observatory in Canada[39]. In addition, they developed hybrid models by combining MARS and RF models with SETAR and GARCH models, and they showed that the RF-SETAR models had the highest accuracy of the hybrid models. Musarat et al. proposed a machine-learning (ML) approach to predicting water levels to prevent the devastation caused by extreme water levels rising in the Kabul River [40]. They used a variety of machine learning models, among used ML model, and they showed that ARIMA model

has the highest performance. Ghimire, S. et al. developed a new AI model based on LSTM and Convolution neural network (CNN) for hourly flow forecasts of the Brisbane River and Teewah Creek in Australia[41]. They designed a prediction model based on six preceding values through statistical self-analysis of the time series from time-series data for the river flow. They also set different time intervals, such as one week, two weeks, four weeks, and nine months. Huang et al. proposed deep-learning models for flow prediction in astrophysics as an alternative to physically based models because the physically based models for flow prediction were relatively slow and costly[42]. Their study used the SOBEK model which is physically based model and three neural network models: ANN, LSTM, and adaptive neural fuzzy inference system (ANFIS). They showed that the LSTM model had the highest accuracy. Debbarma et al. utilized the gamma memory neural network (GMN) and genetic algorithm-gamma GMN (GA-GMN) to predict the daily flow rate. They showed that the GA-GMN model performed better than the GMN model[43]. Senent-Aparicio et al. suggested that instantaneous peak flow (IPF) is very important for reducing flood damage and is combined with ML models and soil and water assessment tool (SWAT) simulations to suggest an approach to estimating IPF[44]. They applied an ANN, ANFIS, support vector machine (SVM), and extreme learning machine (ELM) as the ML models and compared the performance. Studies on river flow prediction using deep learning[39–44] show that the algorithms used to predict river flow are limited in predicting outliers using data-based algorithms. Therefore, we propose the PNP algorithm, which can predict the flow of rivers using climate change-sensitive precipitation.

Table 2
Related river flow prediction using machine-learning

Num	Country	Algorithm	Description
1	Canada	MARS, RF	• This study predicted monthly river flows using three AI approaches: ANN, MARS, and RF. • Among the hybrid models developed, the RF-SETAR models were generally the most accurate, improving the river flow modeling.
2	Afghanistan	ARIMA	• This approach used an ML tool known as an autoregressive integrated moving average for statistical methodological analysis to predict stream flow.
3	Australia	CNN-LSTM	• CNN layers were used to extract the time-series river flow features, while LSTM networks used these features from the CNN for river flow time-series prediction. • River flow prediction was conducted for different time intervals, including one week, two weeks, four weeks, and nine months. • The results showed that the proposed CNN-LSTM model based on the novel framework yielded more accurate predictions. Thus, CNN-LSTM has significant practical value in river flow prediction.
4	China	LSTM, ANFIS, SOBEK	• This work focused on evaluating the reliability of three neural network models (ANN, LSTM, ANFIS) and one physically based model (SOBEK) in terms of efficiency and accuracy for average and peak streamflow simulation. • LSTM models can generally compete with physically based models in flow simulations of complex urban river systems by providing fast flow predictions with acceptable accuracy.
5	India	GMN, GA-GMN	• The authors showed how to predict daily river flows using a memory-based ANN. They chose two different networks: GMN and GA-GMN. • The GA-GMN model gave better results for both datasets. Therefore, it was chosen as an effective tool to predict the flow features of the Dholai River.
6	Spain	ANN, ANFIS, SVM, ELM	• This study proposed an approach to instantaneous peak flow estimation by combining SWAT simulations and ML models. • The SWAT model was used to estimate the maximum average daily flow, and the ML model was used to estimate the instantaneous peak flow based on the maximum average daily flow.
-	Korea	PNP	• In the current paper, we propose an algorithm that can be applied to the water circulatory system and water retention times. • We use not only weather input variables but also upstream operating data as input variables to predict the river flow.

3 Pnp Algorithm

3.1 Characteristics of the PNP algorithm

On Earth, water always flows, and the process of water completing its cycle is called water circulation. This circulation is the continuous flow of water above and below the Earth's surface. It is usually caused by water evaporating from the sea and forming clouds, which transfer the water to land through rain. Once on land, the water flows over the surface of the earth to a lower place on the surface, forming a stream or a river. The stream's water flows downstream into the ocean, completing the water circulatory system. The time when water stays in this circulatory system is called the residence time. The average residence time in Korea is 1.5 weeks in the atmosphere and two weeks in the river. Even in the absence of precipitation, the river flow characteristics require a deep-learning algorithm that can identify the characteristics of the water circulation system to predict the river flow. Therefore, we present the PNP algorithm, a new deep-learning algorithm that considers the water circulation system and the residence time. The deep-learning model consists of input, hidden, and output layers. The input data are fed into the input layer, and the output layer produces the model value. Hidden layers are made up of neurons composed of deep-learning algorithms. The PNP algorithm can consist of neurons in the deep-learning model, such as MLP, LSTM, and other deep-learning algorithms. Figure 2 shows the layers and neurons of the deep-learning model.

The PNP algorithm focused on river flow that we present is divided into water entering the stream (positive water) and water exiting to the sea (negative water). If there is more positive water than negative water, the river becomes too full, which causes flooding, while more negative water than positive water leads to droughts. We applied positive and negative water to the PNP algorithm, corresponding to the positive and negative neurons, respectively, and we also applied features from past precipitation data to consider the residence time.

The PNP algorithm is organized in a hidden deep-learning layer, such as LSTM or RNN, consisting of positive and negative neurons in the hidden layer. Figure 3 shows the PNP algorithm concept. Both neurons receive the precipitation as input. Positive neurons analyze the increasing factor of the water flow in the river, while negative neurons analyze the decreasing factor. There is a line connected between the negative and positive neurons called the conveyor belt, which delivers the results of each neuron to the next node and provides past precipitation information. Through this process, we can reflect the amount of river emissions during the residence time. The conveyor belt provides the calculated value from the previous neuron to the next neuron. This is similar to the cell state in LSTM, which completely removes the cell state's information. However, unlike the cell state, the conveyor belt maintains the information and transfers it to the next neuron. Finally, when the last positive and negative neurons finish calculating, two results are generated from one node of the PNP hidden layer. Figure 4 shows the positive and negative neuron configuration diagram.

The positive and negative neurons are calculated in four steps: input, judgment, application, and transfer. The first step of the positive neuron is the input step, which is given to the neuron and multiplied by the weighted value w. The second step is the judgment by applying the $sigmoid$ function, using the value one if the need is high and zero if it is low. The third step is the application step, which adds the value of the sigmoid function, the positive bias term, b_p, and the value from the previous positive conveyor belt. Finally, in the fourth step, we remove a negative element by applying a rectified linear unit (ReLU) function to the value calculated in the third step. The values removed by the ReLU are transferred to the conveyor belt and the next neuron. The negative neuron follows the same process through the second computational step, but in the third step, b_n is added to the negative bias term rather than the positive bias term, b_p. The fourth step is applied to prevent the quantity value. Equations (1)–(4) show the calculation of positive neurons, while Equations (5)–(8) show the calculation of negative neurons.

$$fs_p = I \times w_p$$

1

$$ss_p = sigmoid\left(a_p\right) \times fs_p$$

2

$$ts_p = ss_p + |b_p| + PCb_{(t-1)}$$

3

$$Ls_p = Relu\left(ts_p\right)$$

4

$$fs_n = input \times w_n$$

5

$$ss_n = sigmoid\left(a_n\right) \times fs_n$$

6

$$ts_n = ss_n + |-b_n| + NCb_{(t-1)}$$

7

$$Ls_n = -Relu(-ts_n)$$

8

In Equations (1)–(8), fs_p and fs_n are the first step in the PNP algorithm computational process, ss_p and ss_n are the second step, ts_p and ts_n are the third step, and Ls_p and Ls_n are the final step. w_p and w_n

are the positive and negative weighted values, respectively. a_p $\mathrm{and} a_n$ are the positive and negative sigmoid functions, respectively. b_p and b_n are the positive and negative bias terms, respectively. I is the input data, $PCb_{(t-1)}$ represents the previous positive conveyor belt value, and $NCb_{(t-1)}$ represents the previous negative conveyor belt value.

3.2 Deep-learning configuration of PNP algorithm

PNP algorithms can be configured as hidden ANN layers, and the number of nodes can be adjusted, as in other deep-learning algorithms such as MLP, LSTM, and GRU. Figure 5 shows the hidden layers and internal nodes comprising the PNP algorithm. The input value of the PNP algorithm is entered in three dimensions, as with the LSTM algorithm. The input data dimensions consist of the data size, variable, and time step. The data size is the amount of data in time units used in the algorithm. The variable represents the number of variables. The time step is the number of past data points used, for which we must consider the residence time of the river flow. We can designate the number of neurons in positive and negative time steps through the time step value. The neuron array is organized in a parallel format based on the number of variables. The neuron array has a two-dimensional structure composed in a series according to the time step. Figure 6 shows the algorithm structure according to the number of variables and neurons.

4 Acquired Data Characteristics

To predict the river flow through deep learning, we must consider the correlation between the target value and the input variables. Because the river flow includes the circulation process of water and the characteristic of flowing from upstream to downstream, we should select input variables that have such a characteristic of water or affected. The water circulation process is largely determined by weather (water quantity, temperature, etc.) and the upstream impact of dams, depending on how the dams operate. Therefore, we collected river flow data, weather observation data closest to the flow measurement point, and upstream dam operation data to predict the river flow. In addition, we obtained datasets from three regions (Hangang Bridge, Yeoju Bridge, and Gangchung Bridge) to generalize the performance evaluation of the algorithms.

4.1 River flow

We acquired data measured at the Hangang, Yeoju, and Gangchung Bridges in Korea. The Hangang Bridge (37°31'03.3"N, 126°57'31.7"E), located over the Han River in Seoul, has a total length of 1,005 m. The Han River is the second longest river in South Korea (494 km) and the largest river in terms of flow and basin. The Han River has hundreds of tributaries, including the Bukhan and Imjin Rivers, across four regions: Gyeonggi, Gwandong, Haeseo, and Hoseo. The Yeoju Bridge (37°29'81.7"N, 127°64'81.4"E) is located at the southern end of the Namhan River from Sangdong to Cheonsong-dong, Yeoju-si, Gyeonggi-do. The Gangchung Bridge (36°82'16.1"N, 126°93'36.6"E), which crosses the Gokgyo river in Yeomchi, is 325 m long and 10 m wide. The blue color in Fig. 7(a) indicates the Han River, and the red checkpoints

indicate the locations of the Hangang and Yeoju Bridges. The blue color in (b) indicates the Gokgyo river, and the red checkpoint indicates the location of the Gangchung Bridge. The Hangang and Yeoju Bridges are located in the same Han River area, but there are differences upstream and downstream, and the Gangchung Bridge is located in a different basin. The Hangang Bridge is the largest by region, followed by the Yeoju Bridge, and finally, the Gangchung Bridge is the smallest.

Table 3

Analysis of river flow (m^3/s) data obtained

Location	Period	Min	Max	Mean	Std.
Hangang Bridge	1/1/2010–12/31/2020	12.78	20680.17	484.188	1113.083
Yeoju Bridge	1/1/2010–12/31/2020	11.06	5493.91	231.759	418.218
Gangchung Bridge	1/1/2010–12/31/2020	4.4	696.22	55.4	41.798

The river flow data were provided by the Han River Flow Control Office for the period 1/1/2010–12/31/2020. The unit of the data is one day and is in the form of 4018 int. Table 3 shows the information of the acquired river flow data, and Fig. 8 shows the obtained river flow graph.

4.2 Weather

As Korea is in the mid-latitudes of the northern hemisphere, changes are evident in all four seasons, with periodic characteristics throughout the year as the seasons change. We used the weather observation data closest to the river flow measurement area and collected data from the Korea Meteorological Administration. Figure 9 shows the locations of the weather station and the flow station, and Table 4 summarizes the weather data.

Table 4
Analysis of weather data acquired

Location	Period	Sort	Min	Max	Mean	Std.
Hangang Bridge	1/1/2010–12/31/2020	Temperature[℃]	−14.8	33.7	13.0	10.8
		Precipitation[m^3/s]	0.0	301.5	3.68	14.401
		Humidity[%]	17.9	99.8	59.7	15.1
Yeoju Bridge	1/1/2010–12/31/2020	Temperature[℃]	−14.8	31.6	11.9	10.7
		Precipitation[m^3/s]	0.0	247.5	3.5	12.9
		Humidity[%]	15.6	100	65.5	15.0
Gangchung Bridge	1/1/2010–12/31/2020	Temperature[℃]	−13.7	30.8	12.3	10.4
		Precipitation[m^3/s]	0.0	232.7	3.3	12.1
		Humidity[%]	16.3	99.3	68.7	13.4

4.3 Upstream dam hydrological data

Dams reduce the peak flood of downstream rivers during flooding and provide water for continuous river maintenance during droughts. They affect downstream changes in spill characteristics in the short term. On the other hand, they also increase flows in the dry season for the long term caused by increased low flows. For this reason, we must use upstream floodgate information as an input variable to predict the flow of the stream. Thus, we collected upstream dam operation data around the flow observation point. We used hydrological data provided by the Korea Water Resources Corporation and Korea Hydro & Nuclear Power for the dams upstream on the Han River. Figure 10 shows the location of the dams on the Han River, and Table 5 summarizes the data from the dam.

There are 10 dams upstream from the Hangang Bridge, three dams upstream from the Yeoju Bridge (the Hoengseong, Chungju, and Goesan Dams), and none upstream from the Gangchung Bridge.

Table 5

Analysis of discharge (m^3/s) data of the dams upstream from the Hangang Bridge

Location	Period	Min	Max	Mean	Std.
Paldang Dam	1/1/2010–12/31/2020	0.0	15870.583	422.778	932.7
Cheongpyeong Dam	1/1/2010–12/31/2020	0.0	9985.75	181.625	436.982
Uiam Dam	1/1/2010–12/31/2020	1.1	9948.75	142.506	344.196
Chuncheon Dam	1/1/2010–12/31/2020	0.0	4118.625	68.696	202.649
Soyanggang Dam	1/1/2010–12/31/2020	2.728	2423.94	62.963	97.428
Hwacheon Dam	1/1/2010–12/31/2020	0.0	2945.458	49.229	202.649
Peace Dam	1/1/2010–12/31/2020	0	2179.22	52.365	114.605
Hoengseong Dam	1/1/2010–12/31/2020	0.036	363.322	4.467	12.463
Chungju Dam	1/1/2010–12/31/2020	0.0	3960.37	136.99	235.694
Goesan Dam	1/1/2010–12/31/2020	0.0	684.042	12.239	34.539

5 Performance Verification Of The Pnp Algorithm

5.1 Algorithm performance verification pipeline configuration

We built a pipeline to verify the algorithm performance. The pipeline sequence is divided primarily into the data and deep-learning processes. Figure 11 indicates the configuration of the algorithm performance verification pipeline. The first step of the data process divides the acquired data into learning and test

data in an 80:20 ratio. The learning data are used to train the deep-learning model, and the test data are used to verify the model performance. We also applied K-fold cross-validation to prevent data with high noise values from being concentrated on one side when dividing the learning and test data and overfitting to a specific dataset. K-fold cross-validation creates K divisions of the learning and test data. In this paper, we organized five types of K-folds, and we also designated the same ratio for the learning and test data. However, we designated different locations for the test data. Figure 12 illustrates the K-fold cross-validation where K = 5.

We organized a preprocessing model for the learning data. The preprocessing model is for standardization, using the learning data divided between learning and validation in a 60:20 ratio. Figure 13 shows the division of the acquired data.

We divided the acquired data into learning and testing data and the learning data into training data and validation data. The learning data were used to construct preprocessing models, and the training data were used to train the models. The validation and testing data were used to measure the performance of the deep-learning model, with the validation data used in constructing or training the model. Validation data were used for preprocessing and training the deep-learning model, for which the testing data were not involved. After applying the training, validation, and test data to the preprocessing model, we built a dataset to fit the input of the deep-learning model. We organized the data building in three dimensions, namely, the data size, variable, and time step, to fit the PNP algorithm input requirements. The time step must be selected considering the residence time in the river. The average residence time in Korean rivers is 14 days; thus, we designated 14 days as the time step. In this study, we designated the variable value as one because the input value to be entered into the PNP algorithm is one of the closest precipitation points to the river flow observation point. The total data size is 3977 data records based on the past data considering the time step. Therefore, the size of the input data of the training, verification, and testing datasets was [2386, 1, 14], [795, 1, 14], [796, 1, 14].

The first stage of the deep-learning process is the model configuration. We constructed the model with the PNP and MLP algorithms, designating 50 nodes for each algorithm. We used the Adam optimizer and mean squared error for error measurement. We constructed another deep-learning model consisting of LSTM algorithms with excellent performance in time-series prediction to compare the performance. This model consisted of LSTM and MLP algorithms, with the number of nodes set to 50. After constructing the model, we used the training and validation data to train it. We input the trained model as test data to predict the river flow and then applied reverse preprocessing. We then measured the error by comparing the reverse-prediction result with the actual river flow value using RMSE and sMAPE for error measurement. We compared the predictive results of the two models with two error-measurement methods to demonstrate the PNP algorithm performance. Eq. (9) represents the preprocessing, and Eq. (10) represents the reverse preprocessing. Eq. (11) represents RMSE, and Eq. (12) represents sMAPE. Figure 14 shows the configuration of the model, with Fig. 14(a) representing the model using the PNP algorithm and Fig. 14(b) showing the model using the LSTM algorithm.

$$Preprocess = \frac{ID - LD_{mean}}{LD_{std}}$$

9

$$Reversepreprocess = PD \times LD_{std} + LD_{mean}$$

10

$$RMSE = \sum \sqrt{(PD - RD)^2}$$

11

$$sMAPE = \frac{|PD - RD|}{(|PD| + |RD|)/2} \times 100$$

12

In Equations (9)–(12), ID is the training, verification, and test data, while LD_{mean} represents the average value of the learning data and LD_{std} represents the standard deviation. PD indicates the prediction value of the model using the test data, and RD indicates the river flow data.

5.2 Verification results

Table 6 gives the RMSE and sMAPE prediction results for the two models. E_1, E_2, E_3, E_4, E_5 represents the K-fold cross-validation in Fig. 10, respectively, while E_{mean} represents the average value of E_1, E_2, E_3, E_4, E_5. The error of the PNP model was lower than that of LSTM for the Hangang, Yeoju, and Gangchung Bridges. Thus, we proved that the PNP algorithm performed better than the LSTM algorithm in predicting the river flow using precipitation.

Table 6
Comparison of model prediction errors

Location	K-fold	RMSE		sMAPE	
		PNP	**LSTM**	**PNP**	**LSTM**
Hangang Bridge	E_1	332.934	367.537	24.979	25.555
	E_2	217.206	310.295	26.461	29.886
	E_3	239.045	317.881	30.888	32.239
	E_4	291.571	328.284	23.830	27.781
	E_5	431.208	477.596	25.617	29.137
	E_{mean}	**302.392**	360.319	**26.355**	28.920
Yeoju Bridge	E_1	136.686	166.380	25.686	26.232
	E_2	141.359	164.857	30.581	33.161
	E_3	95.066	136.162	26.861	34.055
	E_4	143.296	162.093	26.860	29.384
	E_5	174.871	294.072	24.579	26.889
	E_{mean}	**138.256**	184.713	**26.913**	29.944
Gangchung Bridge	E_1	31.457	47.717	32.814	41.450
	E_2	33.050	49.687	30.737	40.433
	E_3	26.550	47.430	24.989	37.393
	E_4	28.640	46.968	26.632	28.030

E_5	34.306	58.868	29.663	42.033
E_{mean}	**30.801**	50.134	**28.967**	37.868

6 Prediction Of River Flow

In this section, we constructed a prediction model of the river flow using temperature, humidity, and upstream dam hydrological data as well as precipitation. We composed two models to compare the performance, as shown in Fig. 15, including the PNP (a) and LSTM (b) algorithms.

Model (a) in Fig. 15 uses the PNP algorithm, which consists of three input layers: precipitation, weather (excluding precipitation), and upstream dam hydrological data. The PNP uses the hidden layer connected from the input precipitation layer, while the LSTM uses the hidden layer connected to the weather and upstream dam hydrological data input layers. Because the PNP algorithm is based on the correlation between precipitation and river flow, we do not use the PNP algorithm for the hidden layers of weather and upstream dam hydrological data. There are 10 dams upstream of the Hangang Bridge and three upstream of the Yeoju bridge. There is no upstream dam input layer for the Gangchung Bridge because it has no upstream dam. We used the PNP algorithm for the hidden precipitation layers, while we used the LSTM algorithm for the hidden layers of the remaining weather (excluding precipitation) and operational information regarding the upstream dams. Finally, we organized a deep-learning model with two hidden MLP layers and one output layer. The model in Fig. 14 (b) used the LSTM algorithm rather than PNP, and the remainder of the deep-learning configuration was identical to the model in Fig. 14 (a). We divided the data into a 60:20:20 ratio for training, verification, and test data. Figure 15 shows the flow data from the Hangang, Yeoju, and Gangchung Bridges into training, verification, and test data. Table 7 represents the output shape of the deep-learning layer of the models in Figs. 15 (a) and (b).

Table 7
Type of input data and output shape of the model

Location	Input layer	Variable of input data	Data shape
Hangang Bridge	Precipitation	precipitation	(4004, 1, 14)
	Weather	Temperature, humidity	(4004, 2, 14)
	Upstream dam operation	Paldang dam, Cheongpyeong, and others	(4004, 10, 14)
Yeoju Bridge	Precipitation	precipitation	(4004, 1, 14)
	Weather	Temperature, humidity	(4004, 2, 14)
	Upstream dam operation	Geosan dam, Chungju dam, Hoengseong dam	(4004, 3, 14)
Gangchung Bridge	Precipitation	precipitation	(4004, 1, 14)
	Weather	Temperature, humidity	(4004, 2, 14)

Table 8 shows the model's prediction error, and Figs. 17 and 18 show the model's prediction and flow measurement.

Table 8
Prediction error of the model

Location	RMSE		sMAPE	
	PNP	LSTM	PNP	LSTM
Hangang Bridge	**198.106**	216.458	**22.121**	22.251
Yeoju Bridge	**87.340**	93.465	**16.173**	16.801
Gangchung Bridge	**27.888**	28.270	**31.504**	31.539

As a result, we concluded that the PNP model performs better than LSTM, according to both RMSE and sMAPE.

7 Conclusion

This paper presented a PNP algorithm based on an understanding of hydrology to predict river flow. We developed the PNP algorithm considering precipitation with the river flow, the length of water residence time, and other variables. The PNP algorithm consists of positive and negative neurons that consider the water circulatory system. We applied a conveyor belt, which uses past data to consider the residence time in the river. The input value of the PNP algorithm is entered in three dimensions, as with LSTM, and the dimensions of the input data are the data size, variable, and time step. We acquired input and output data to train the deep-learning model. The output is the river flow data measured at the Hangang Bridge, Yeoju

Bridge, and Gangchung Bridge, and the input data uses the weather data from the near river flow measurement point and the operational data of upstream dams. We also used K-fold cross-validation to measure the model performance accurately. We constructed two models to verify the performance of the PNP algorithm. One was a model composed of PNP algorithms, and the other was a model composed of LSTM algorithms. We compared the average values by measuring the performance of the models with RMSE and sMAPE with five datasets generated by K-fold cross-validation. We measured the prediction errors in both models: for the Hangang Bridge, the RMSE and sMAPE were 57.927 and 2.565, respectively; for the Yeoju Bridge, the RMSE and sMAPE were 46.457 and 3.031, respectively; and for the Gangchung Bridge, the RMSE and sMAPE were 19.333 and 8.901, respectively. We confirmed the overall high predictive performance of models using the PNP algorithm. Figure 19 is a predictive error comparison graph of the two models from Section 5, where (a) shows the RMSE and (b) shows the sMAPE. We also constructed a river flow prediction model using not only precipitation but also temperature, humidity, and upstream dam hydrological data. To verify the performance of the model, we measured the river flow prediction performance by constructing a model using the LSTM algorithm and measured the prediction errors in the two models: for the Hangang Bridge, the RMSE and sMAPE were 18.352 and 0.13, respectively; for the Yeoju Bridge, the RMSE and sMAPE were 6.125 and 0.628, respectively, and for the Gangchung Bridge, the RMSE and sMAPE were 0.382 and 0.035, respectively. We confirmed that overall the prediction error of the model using the PNP algorithm was low. Figure 20 shows a predictive error comparison graph of the two models from Section 6, where (a) shows the RMSE and (b) shows the sMAPE. In this paper, we suggested a PNP algorithm considering the correlation between precipitation and river flow. We improved the flow rate prediction performance by using the PNP algorithm. However, we never used a reasonable algorithm for the correlation between other input variables (weather, upstream dam data) and the river flow. In the future, we intend to develop improved algorithms that match the input variables and correlations to improve the river flow prediction performance further. Through this, we expect that the prediction of river flow will improve significantly. As a result, we can reduce damage from the natural disasters of drought and floods through river flow prediction.

Declarations

Compliance with ethical standards:

Funding: This work was supported by the Valve Center from the Regional Innovation Center (RIC, B0010565) Program of Ministry of Trade, Industry & Energy (MOTIE).

Conflict of Interest: The authors certify that there is no conflict of interest with any individual or organization for the present work.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors

Authorship contributions

G.B. and Y.B. conceived and designed the conceptualization, G.B. performed the computer simulation and Y.B. analyzed the data and algorithms; G.B. wrote the paper. All authors have given approval to the final version of the manuscript.

References

1. Ren, J., and Xu. L.: On vectorization of deep convolutional neural networks for vision tasks. Proceedings of the AAAI Conference on Artificial Intelligence. 29(1), 1840-1846(2015).
2. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition. 3642-3649(2012).
3. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 25. 1-9(2012).
4. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In Artificial intelligence and statistics. PMLR. 127-135(2012).
5. Cireşan, D. C., Meier, U., Schmidhuber, J.: Transfer learning for Latin and Chinese characters with deep neural networks. In The 2012 international joint conference on neural networks (IJCNN). IEEE. 1-6(2012).
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26. 1-9(2013).
7. Hadsell, R., Erkan, A., Sermanet, P., Scoffier, M., Muller, U., LeCun, Y.: Deep belief net learning in a long-range vision system for autonomous off-road driving. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 628-633(2008).
8. Asgari Taghanaki, S., Abhishek, K., Cohen, J. P., Cohen-Adad, J., Hamarneh, G.: Deep semantic segmentation of natural and medical images: a review. Artificial Intelligence Review, 54(1), 137-178(2021).
9. Tunyasuvunakool, K., Adler, J., Wu, Z., Green, T., Zielinski, M., Žídek, A., ... , Hassabis, D.: Highly accurate protein structure prediction for the human proteome. Nature, 596(7873), 590-596(2021).
10. Quang, D., Chen, Y., Xie, X.: DANN: a deep learning approach for annotating the pathogenicity of genetic variants. Bioinformatics, 31(5), 761-763(2015).
11. Wang, H., Cimen, E., Singh, N., Buckler, E.: Deep learning for plant genomics and crop improvement. Current opinion in plant biology, 54, 34-41(2020).
12. Jaganathan, K., Panagiotopoulou, S. K., McRae, J. F., Darbandi, S. F., Knowles, D., Li, Y. I., ..., Farh, K. K. H.: Predicting splicing from primary sequence with deep learning. Cell, 176(3), 535-548(2019).
13. Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ..., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499(2016).
14. Bai, S., Kolter, J. Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018).

15. Borovykh, A., Bohte, S., Oosterlee, C. W.: Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv:1703.04691(2017).
16. Mudassir, M., Bennbaia, S., Unal, D., Hammoudeh, M.: Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. Neural computing and applications, 1-15(2020).
17. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926(2017).
18. Mudelsee, M.: Trend analysis of climate time series: A review of methods. Earth-science reviews, 190, 310-322(2019).
19. Gharaei, R. H., Sharify, R., Nezamabadi-Pour, H.: An efficient outlier detection method based on distance ratio of k-nearest neighbors. In 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), IEEE, 1-5(2022).
20. Hendrycks, D., Mazeika, M., & Dietterich, T.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606(2018).
21. Pang, G., Shen, C., Cao, L., Hengel, A. V. D.: Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 54(2), 1-38(2021).
22. Field, C. B., Barros, V., Stocker, T. F., Dahe, Q. (Eds.).: Managing the risks of extreme events and disasters to advance climate change adaptation: special report of the intergovernmental panel on climate change. Cambridge University Press, New York(2012).
23. Palmer, T. N., Räisänen, J.: Quantifying the risk of extreme seasonal precipitation events in a changing climate. Nature, 415(6871), 512-514(2002).
24. Lauer, S., Sanderson, M. R., Manning, D. T., Suter, J. F., Hrozencik, R. A., Guerrero, B., Golden, B.: Values and groundwater management in the Ogallala Aquifer region. Journal of Soil and Water Conservation, 73(5), 593-600(2018).
25. Shankar, P. V., Kulkarni, H., Krishnan, S.: India's groundwater challenge and the way forward. Economic and political Weekly, 37-45(2011).
26. Malyan, S. K., Singh, R., Rawat, M., Kumar, M., Pugazhendhi, A., Kumar, A., ..., Kumar, S. S.: An overview of carcinogenic pollutants in groundwater of India. Biocatalysis and Agricultural Biotechnology, 21, 101288(2019).
27. Abidin, H. Z., Djaja, R., Darmawan, D., Hadi, S., Akbar, A., Rajiyowiryono, H., ...,Subarya, C.: Land subsidence of Jakarta (Indonesia) and its geodetic monitoring system. Natural Hazards, 23(2), 365-387(2001).
28. Fowler, A. M., Hennessy, K. J.: Potential impacts of global warming on the frequency and magnitude of heavy precipitation. Natural Hazards, 11(3), 283-303(1995).
29. Papalexiou, S. M., Montanari, A.: Global and regional increase of precipitation extremes under global warming. Water Resources Research, 55(6), 4901-4914 (2019).

30. Myhre, G., Alterskjær, K., Stjern, C. W., Hodnebrog, Ø., Marelle, L., Samset, B. H., ..., Stohl, A.: Frequency of extreme precipitation increases extensively with event rareness under global warming. Scientific reports, 9(1), 1-10 (2019).
31. Hoegh-Guldberg, O., Jacob, D., Bindi, M., Brown, S., Camilloni, I., Diedhiou, A., ... & Zougmoré, R. B.: Impacts of 1.5 C global warming on natural and human systems. Global warming of 1.5° C. (2018).
32. Hubel, D. H., Wiesel, T. N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1), 106(1962).
33. Hubel, D. H., & Wiesel, T. N.: Receptive fields of single neurones in the cat's striate cortex. The Journal of physiology, 148(3), 574(1959).
34. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386(1958).
35. Elman, J. L.: Finding structure in time. Cognitive science, 14(2), 179-211(1990).
36. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, 9(8), 1735-1780(1997).
37. Schuster, M., Paliwal, K. K.: Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681 (1997).
38. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555(2014).
39. Fathian, F., Mehdizadeh, S., Sales, A. K., Safari, M. J. S.: Hybrid models to improve the monthly river flow prediction: Integrating artificial intelligence and non-linear time series models. Journal of Hydrology, 575, 1200-1213(2019).
40. Musarat, M. A., Alaloul, W. S., Rabbani, M. B. A., Ali, M., Altaf, M., Fediuk, R., ..., Farooq, W.: Kabul river flow prediction using automated ARIMA forecasting: A machine learning approach. Sustainability, 13(19), 10720(2021).
41. Ghimire, S., Yaseen, Z. M., Farooque, A. A., Deo, R. C., Zhang, J., Tao, X.: Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks. Scientific Reports, 11(1), 1-26(2021).
42. Huang, X., Li, Y., Tian, Z., Ye, Q., Ke, Q., Fan, D., ..., Liu, J.: Evaluation of short-term streamflow prediction methods in Urban river basins. Physics and Chemistry of the Earth, Parts A/B/C, 123, 103027(2021).
43. Debbarma, S., Choudhury, P.: River flow prediction with memory-based artificial neural networks: a case study of the Dholai river basin. International Journal of Advanced Intelligence Paradigms, 15(1), 51-62(2020).
44. Senent-Aparicio, J., Jimeno-Sáez, P., Bueno-Crespo, A., Pérez-Sánchez, J., Pulido-Velázquez, D.: Coupling machine-learning techniques with SWAT model for instantaneous peak flow prediction. Biosystems engineering, 177, 67-77(2019).

Figures

Image not available with this version

Figure 1

Legend not included with this version

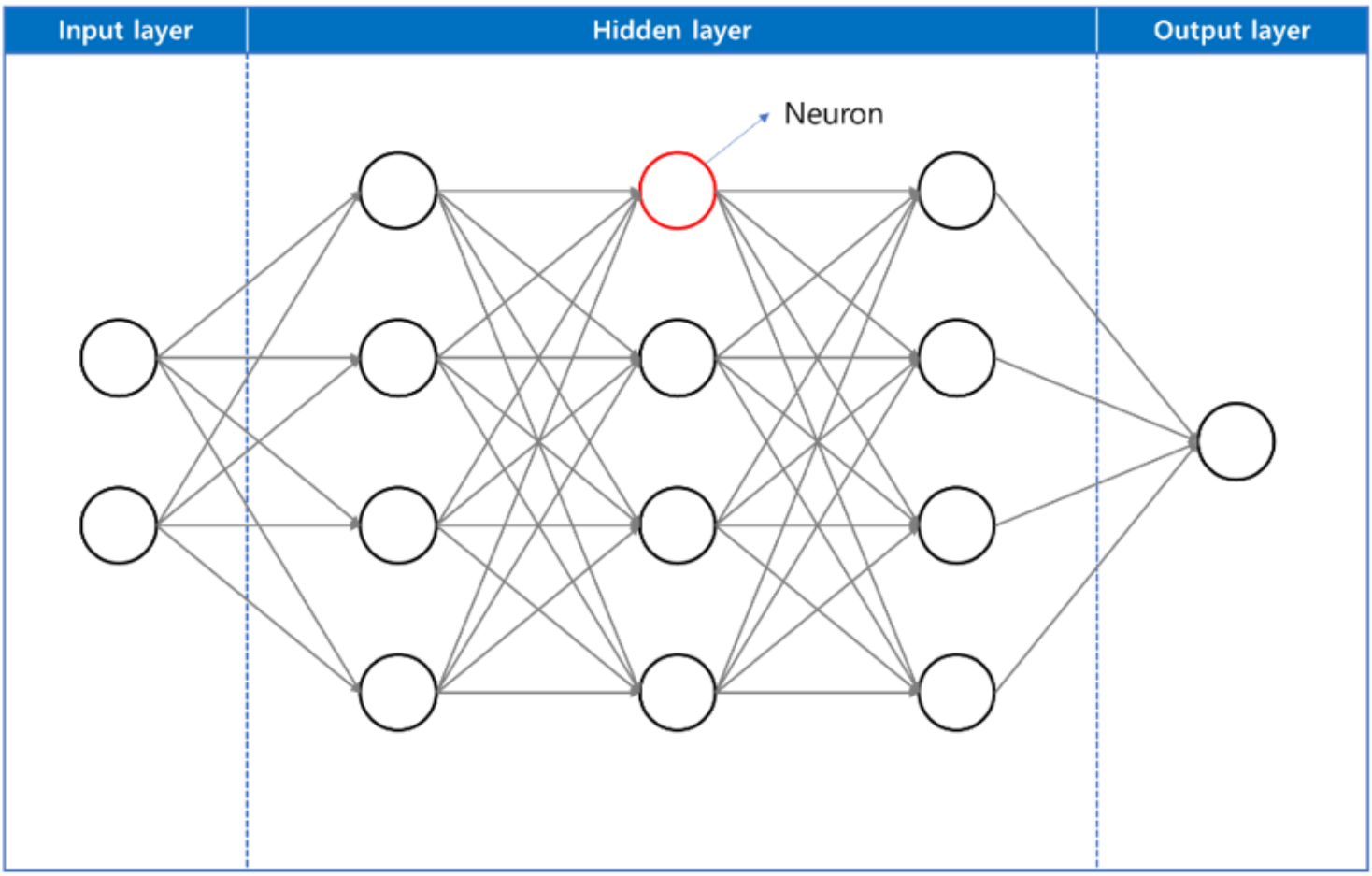


Figure 2

Layers and neurons of the deep-learning model

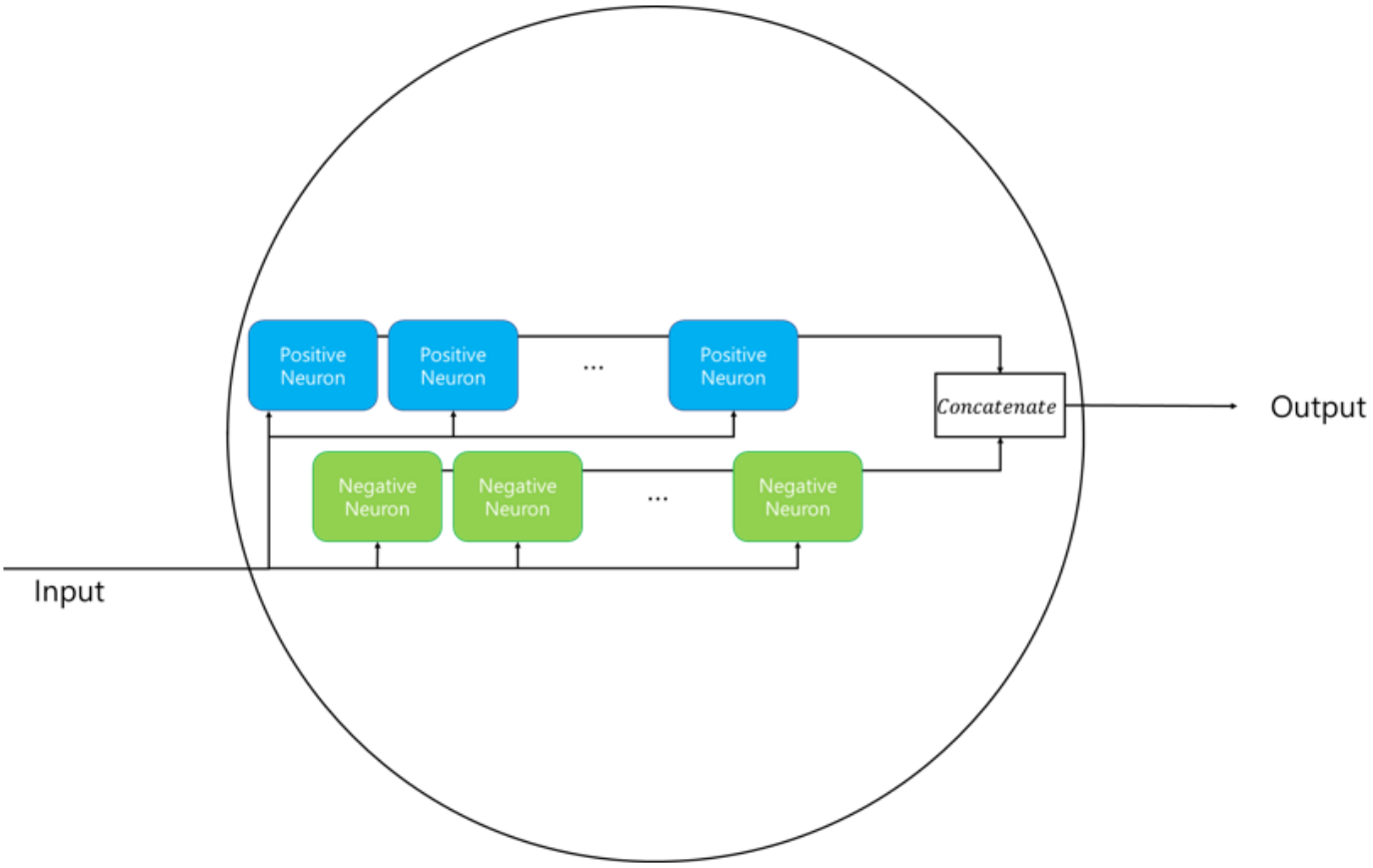


Figure 3

PNP algorithm concept

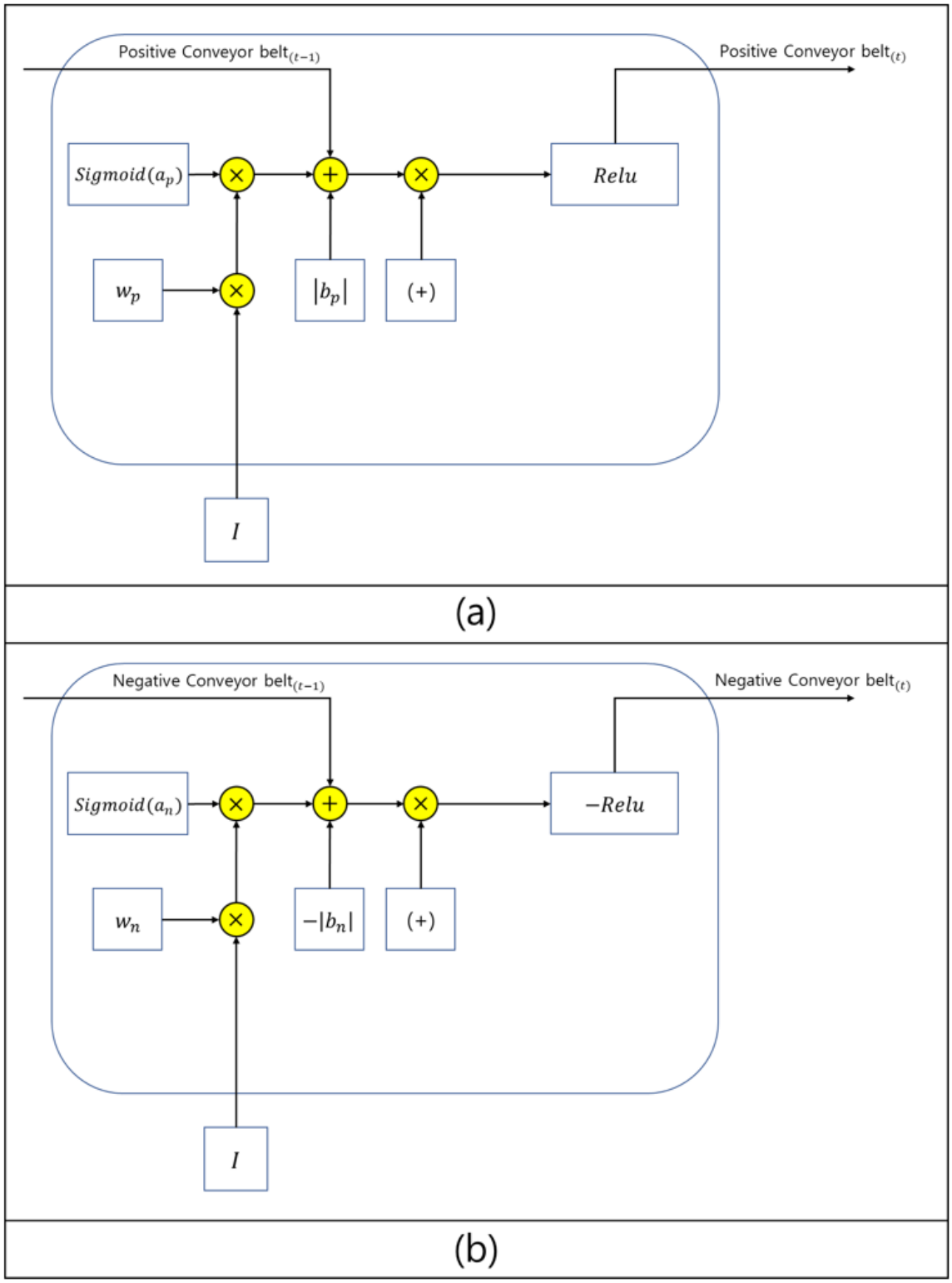


Figure 4

The positive and negative neuron configuration diagram

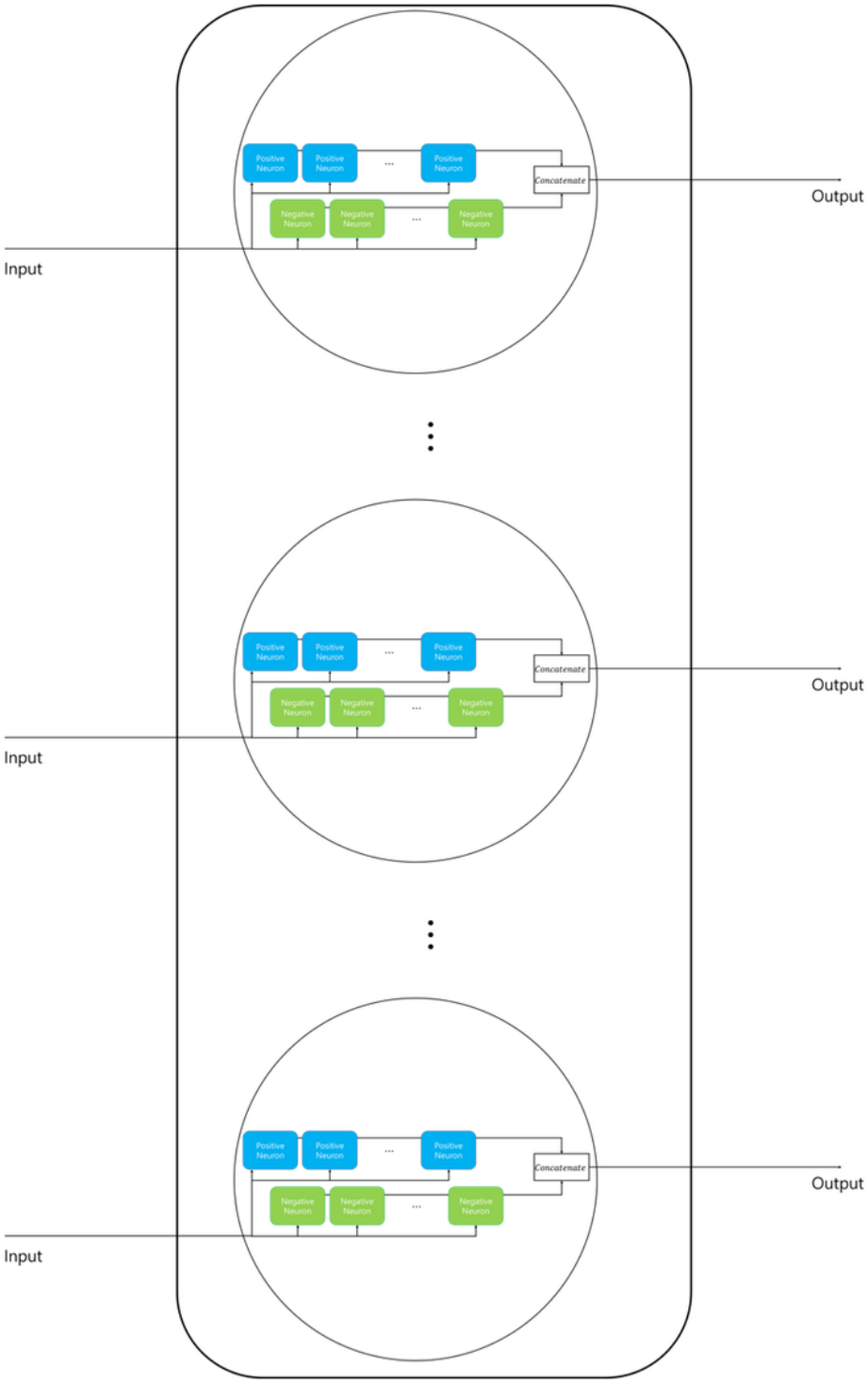


Figure 5

Hidden layers and internal nodes comprising the PNP algorithm

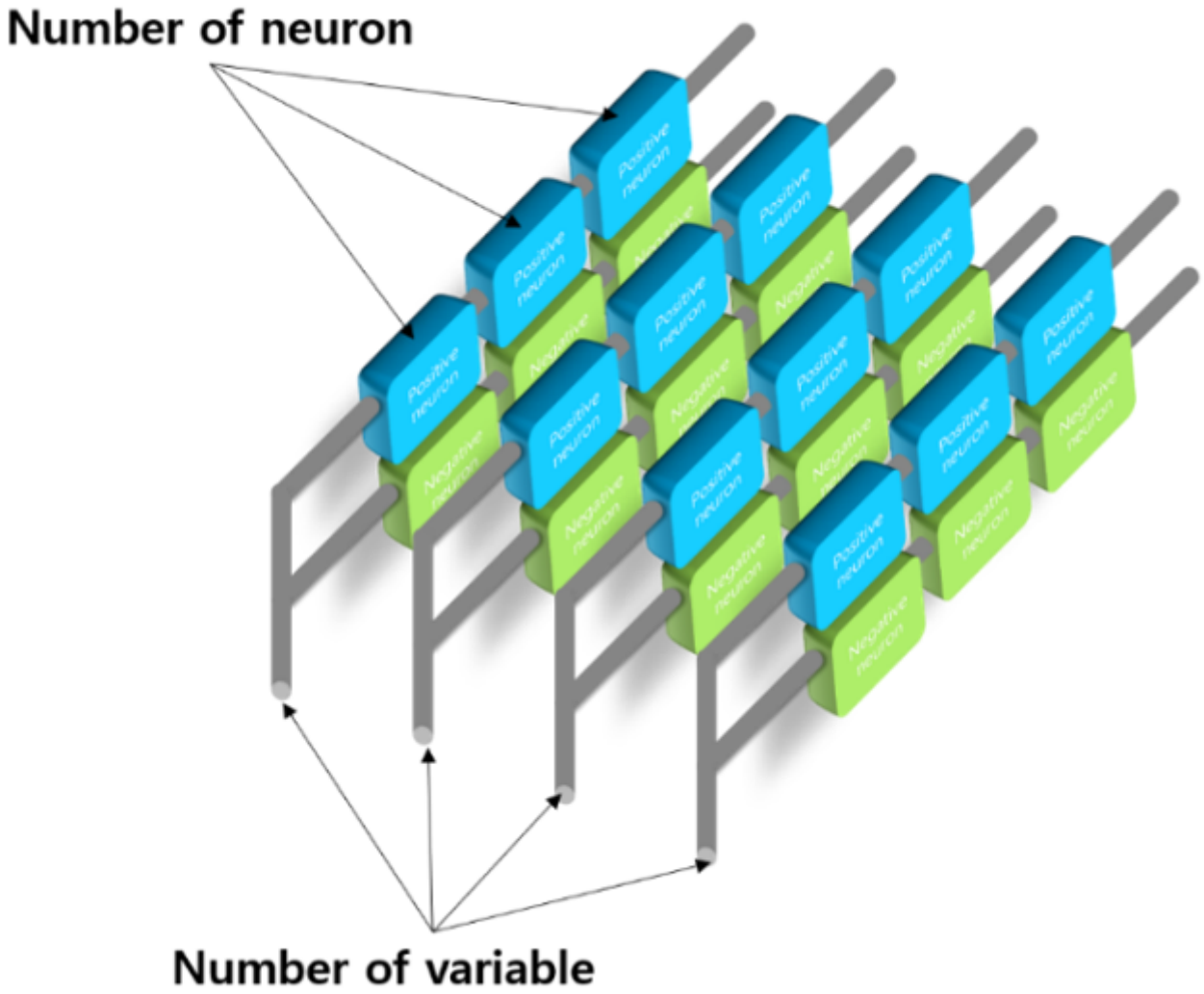
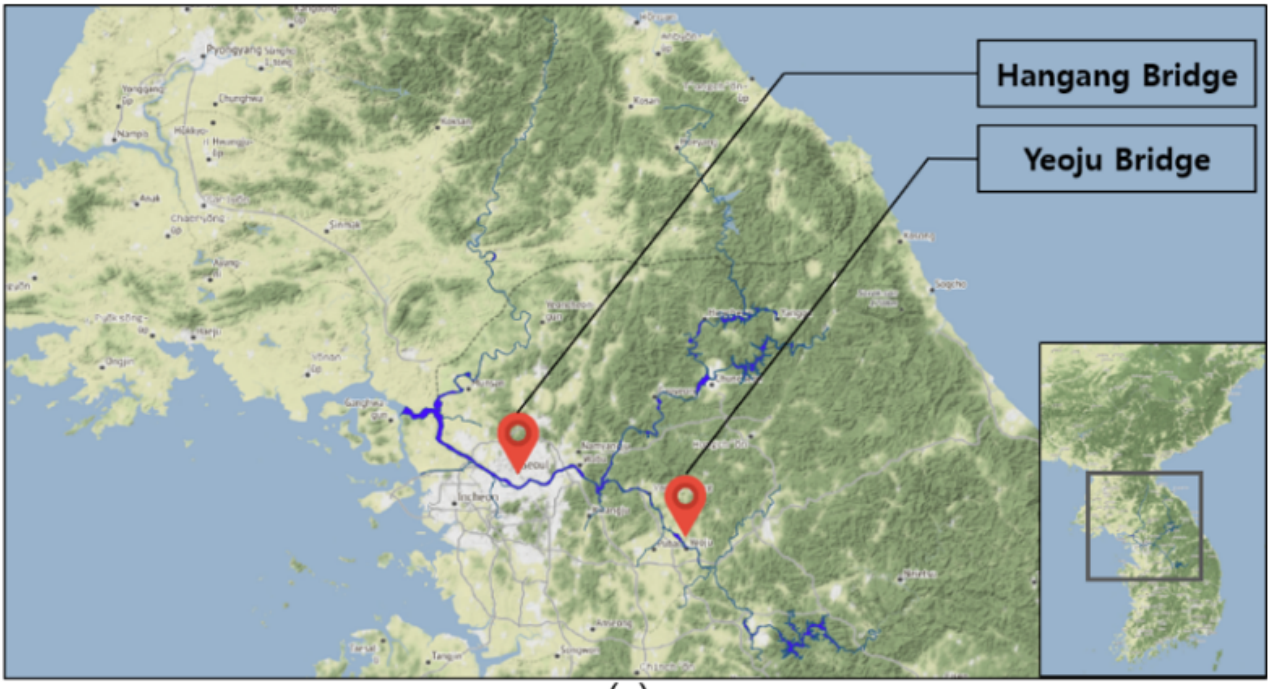
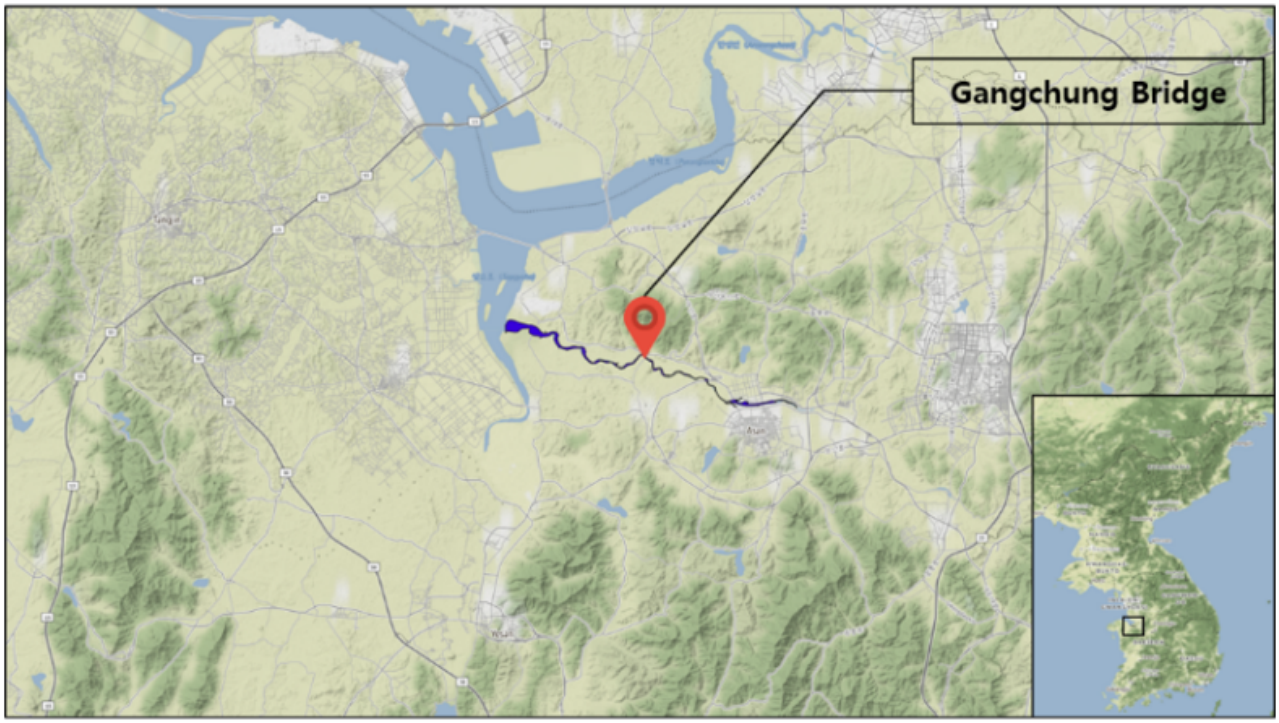


Figure 6

Algorithm structure according to the numbers of variables and neurons



(a)



(b)

Figure 7

(a) Map of the Han River basin and location of the Hangang and Yeoju Bridges; (b) map of the Gokgyo River basin and location of the Gangchung Bridge

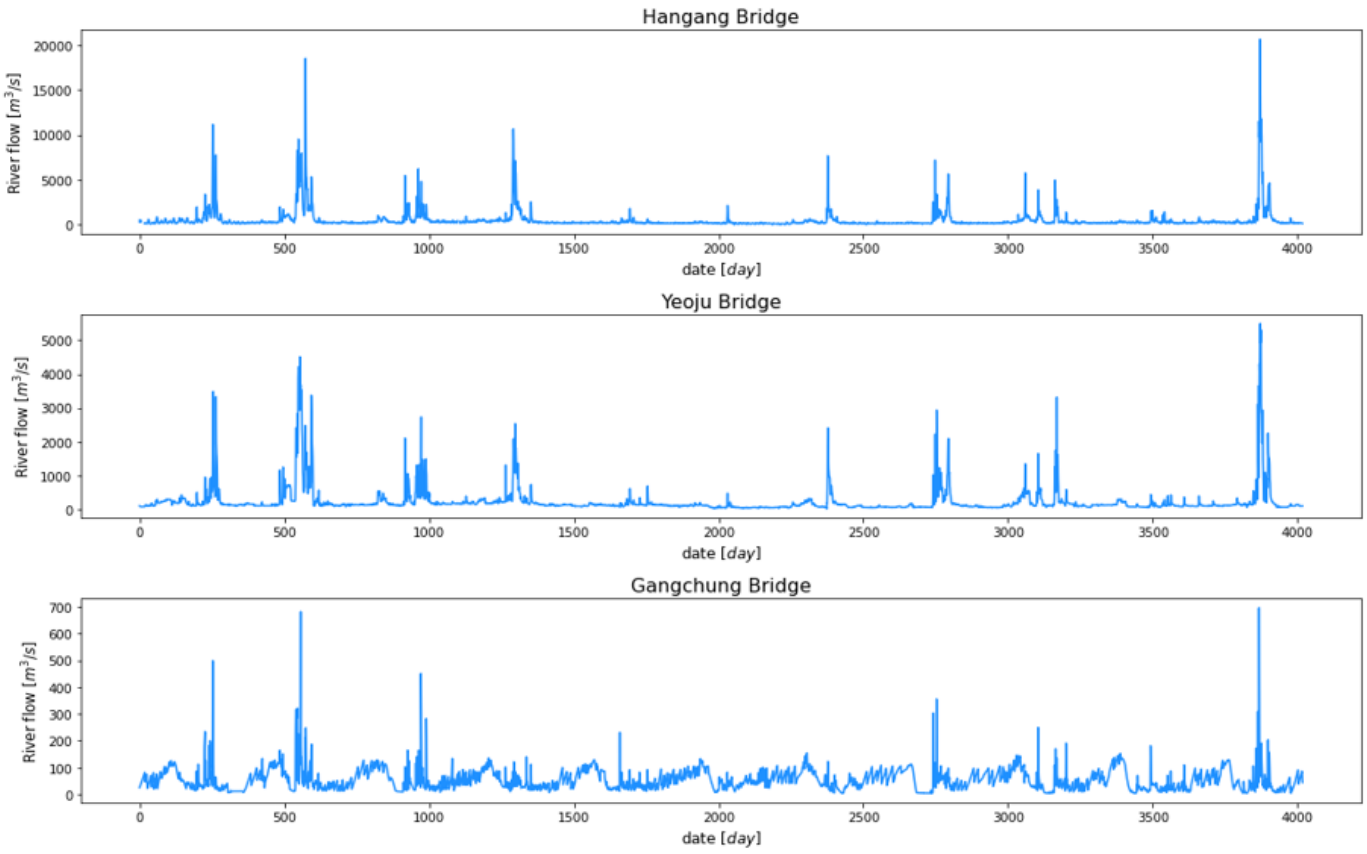


Figure 8

River flow graph (2010–2020)

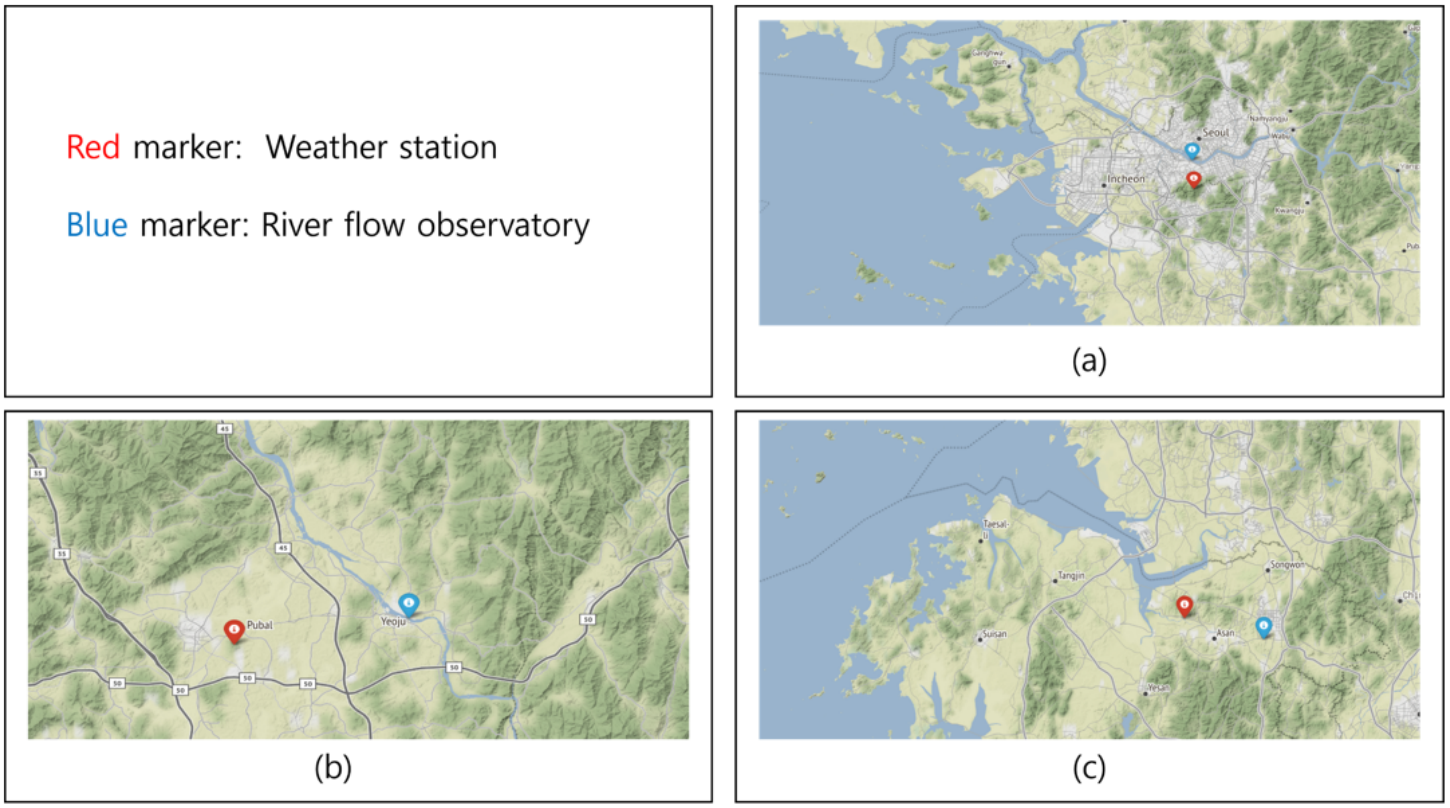


Figure 9

Locations of river flow observatories and weather stations: (a) Hangang Bridge, (b) Yeoju Bridge, and (c) Gangchung Bridge

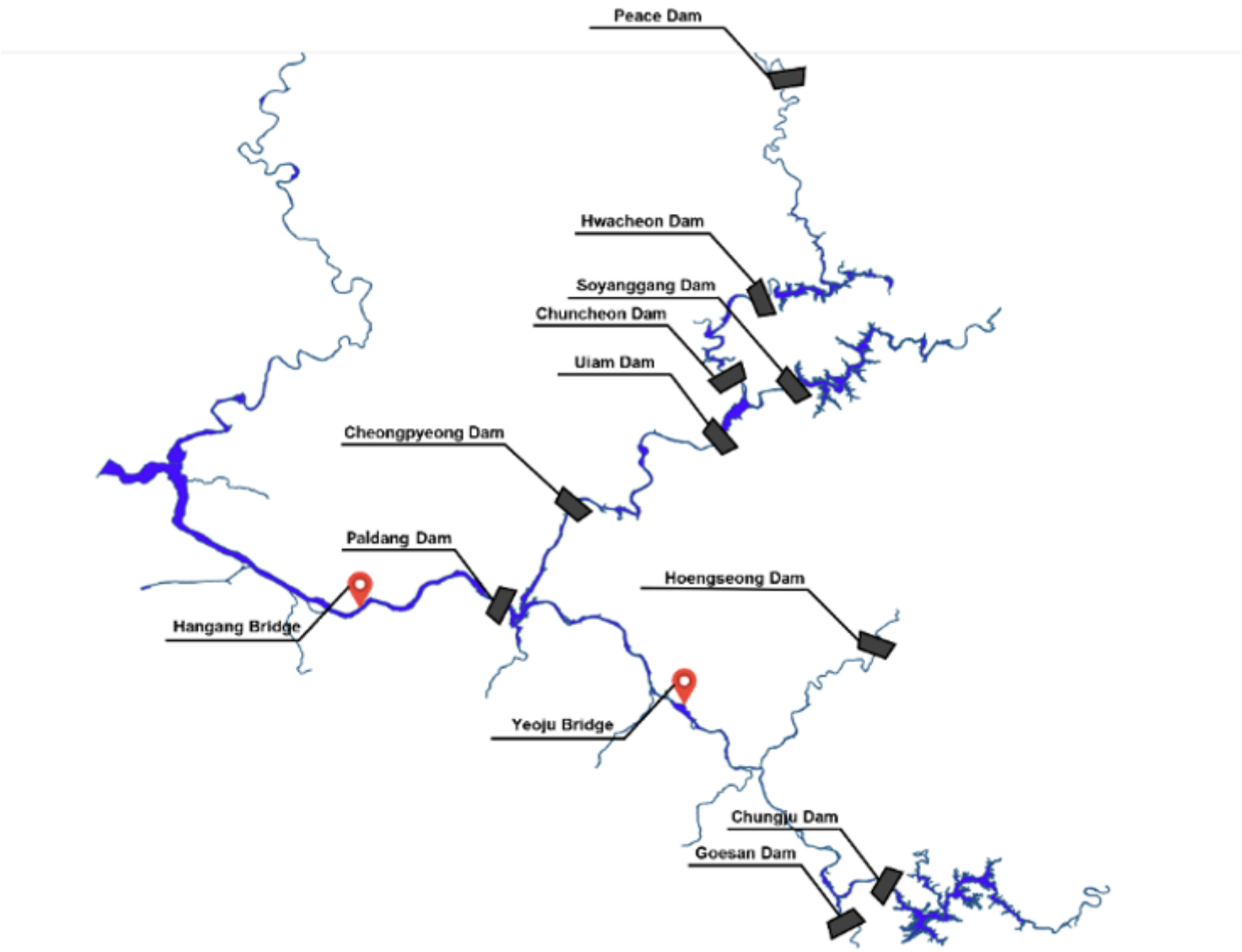


Figure 10

The location of the dams in the Han River basin

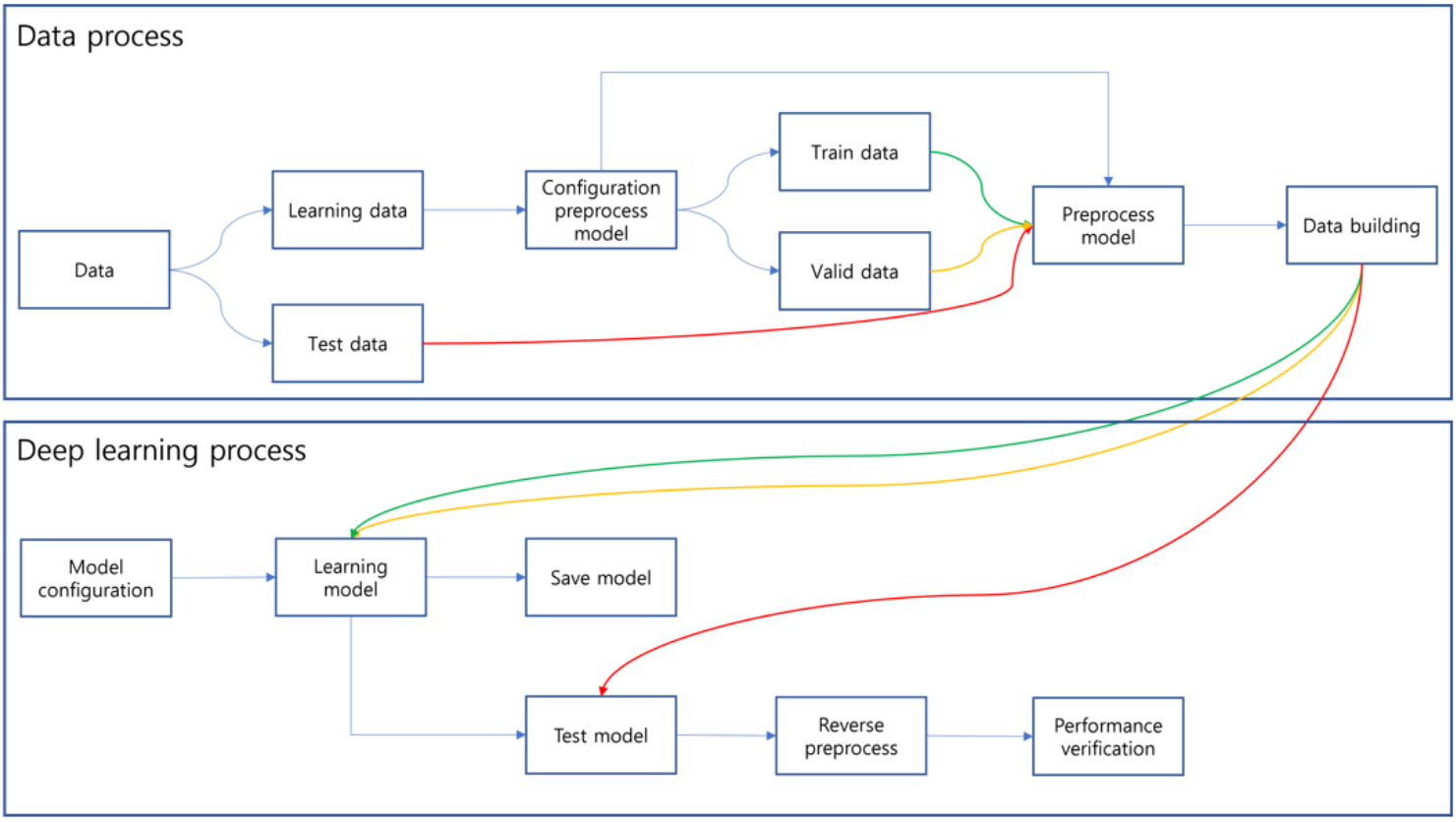


Figure 11

Performance verification of the algorithm pipeline configuration

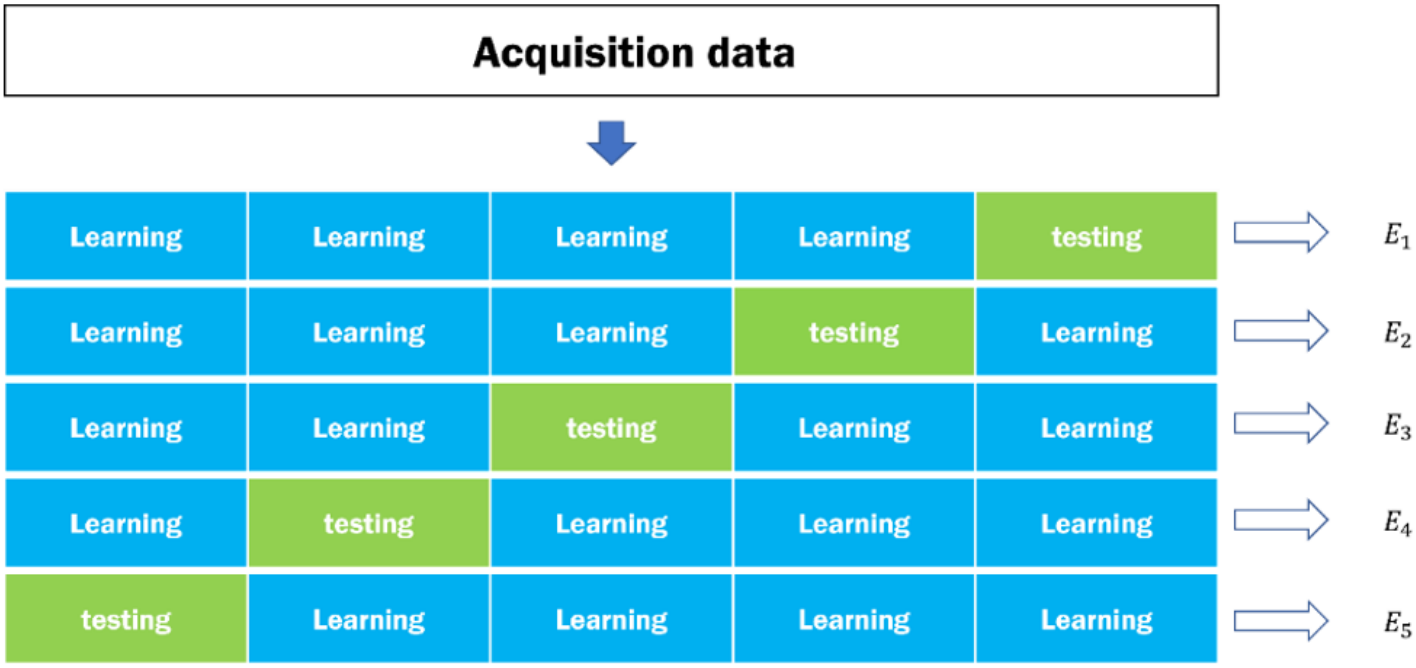


Figure 12

K-Fold cross-validation where K = 5

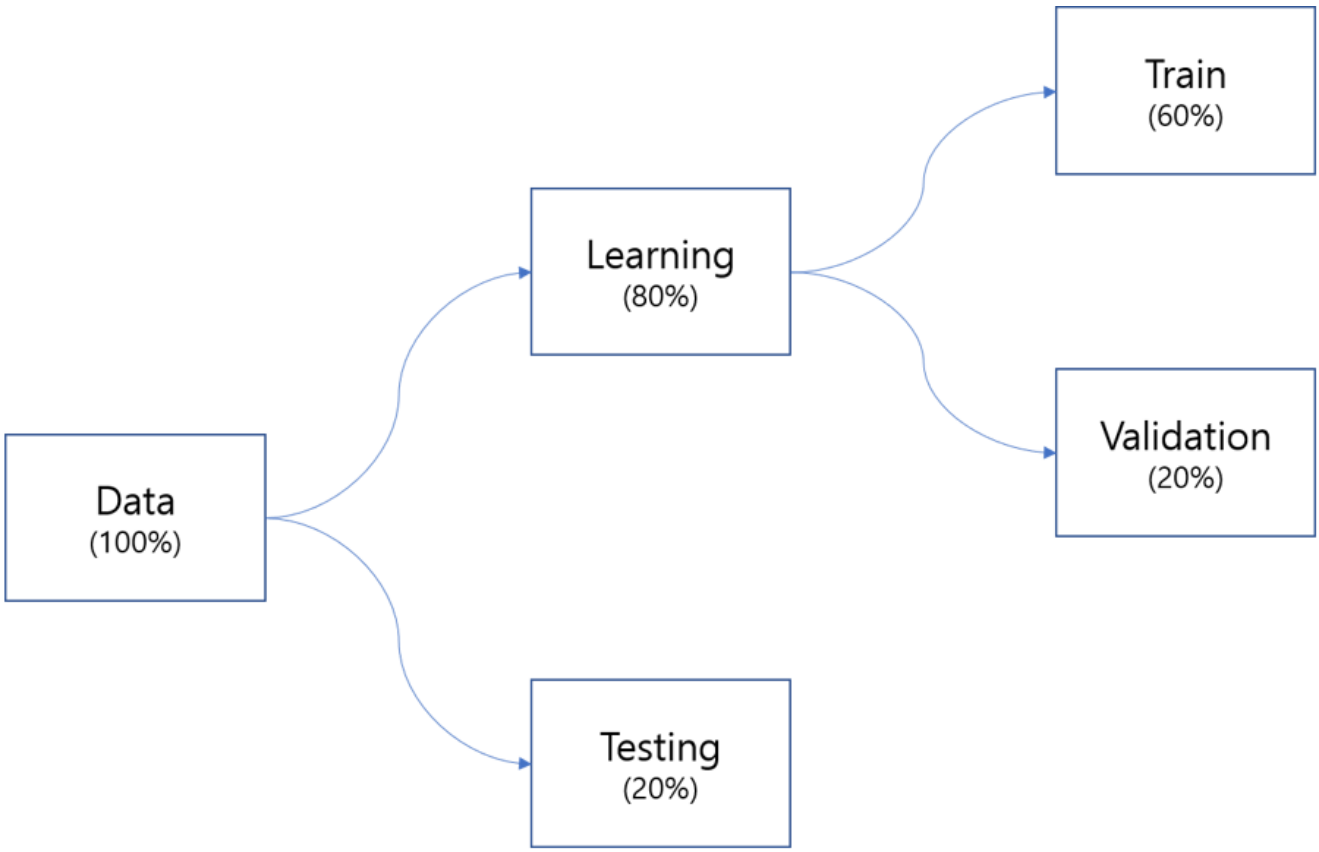


Figure 13

Division of acquired data

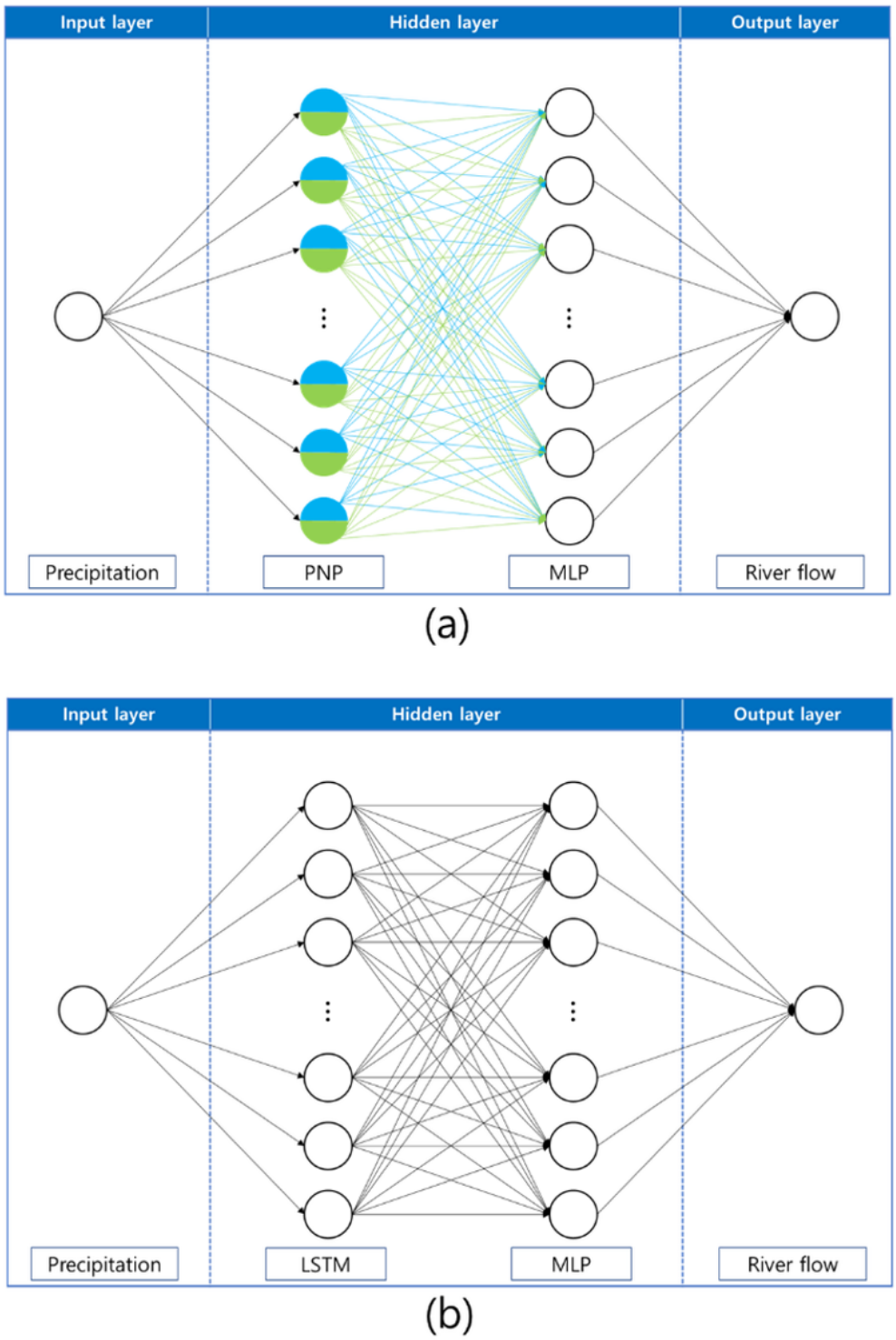
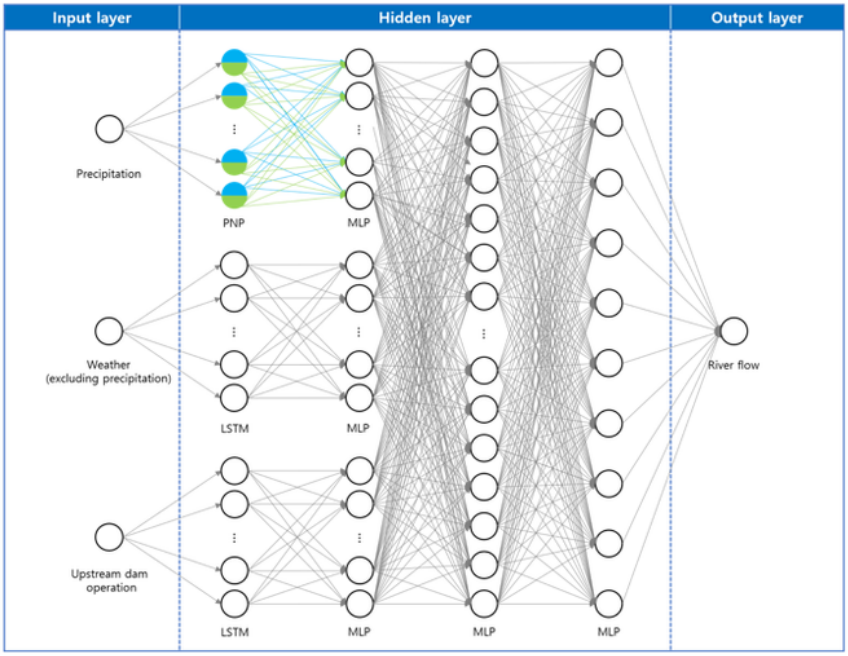
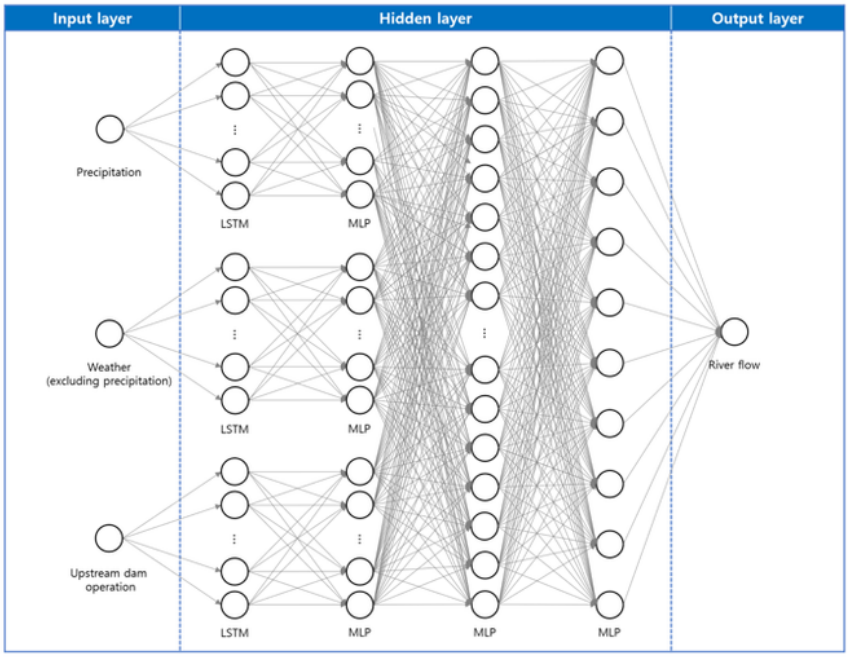


Figure 14

Model composition using (a) the PNP algorithm and (b) the LSTM algorithm



(a)



(b)

Figure 15

The composition of the two models

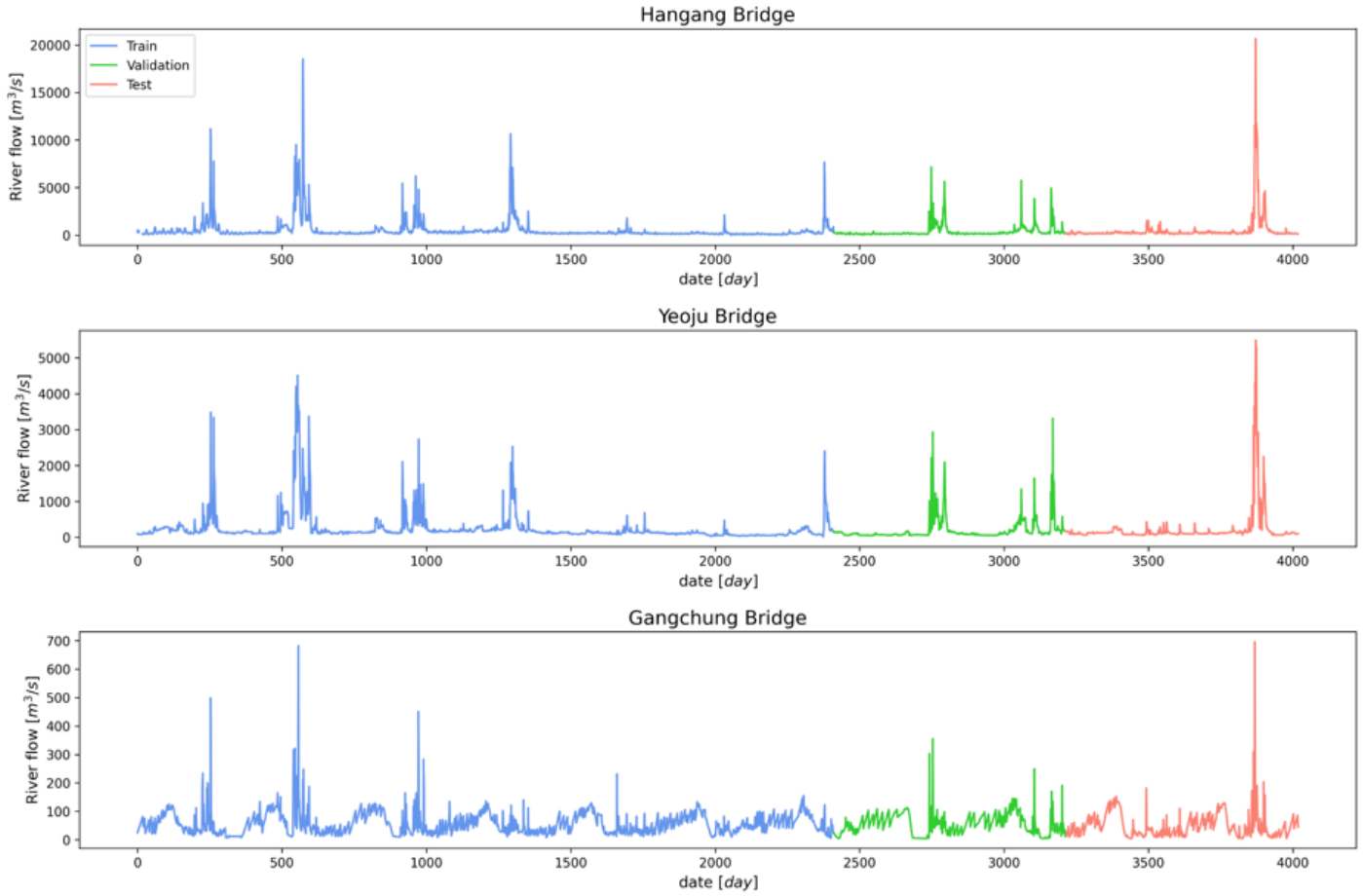


Figure 16

Training, verification, and testing precipitation data

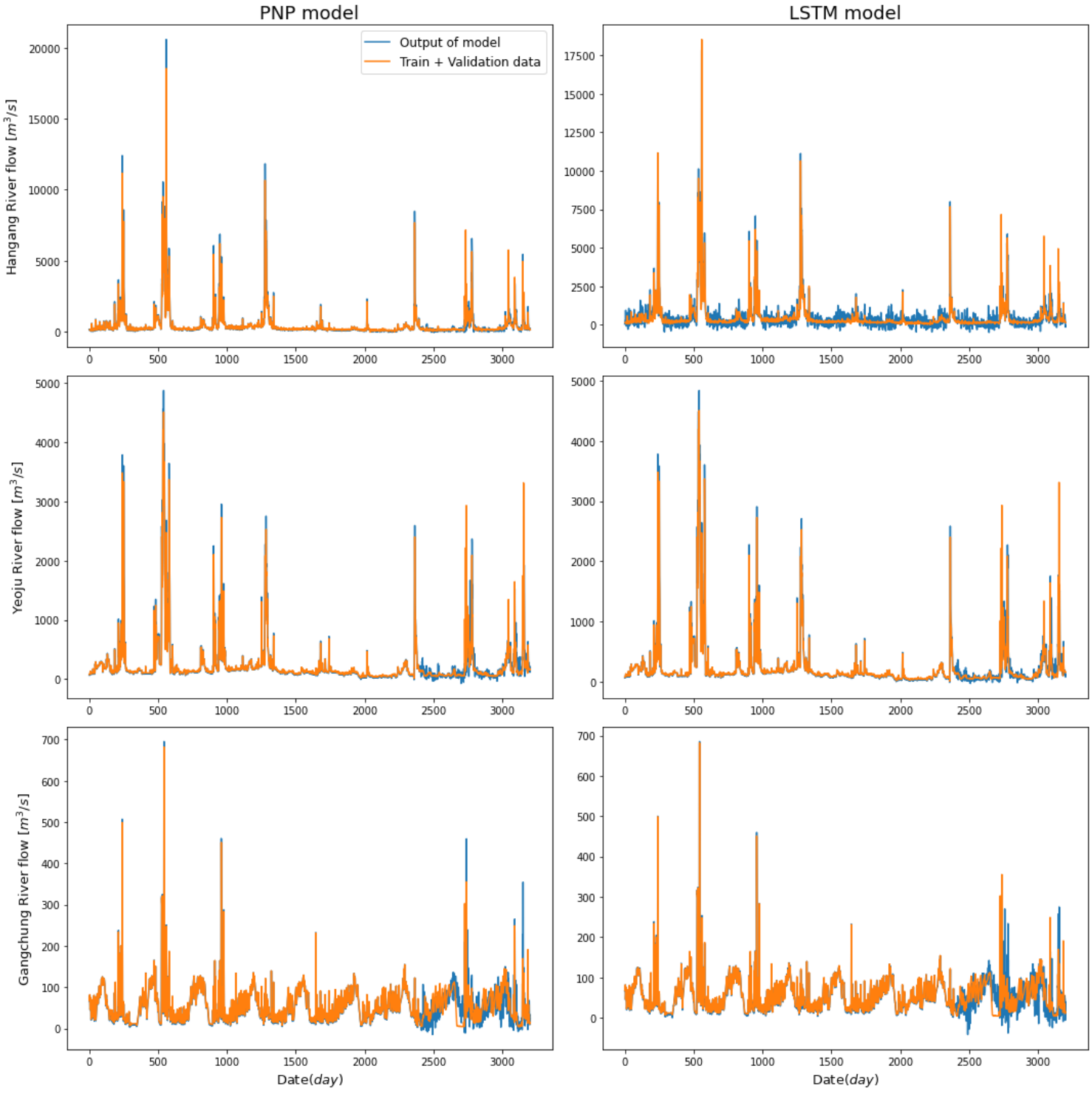


Figure 17

Prediction and observation values of the training and validation data

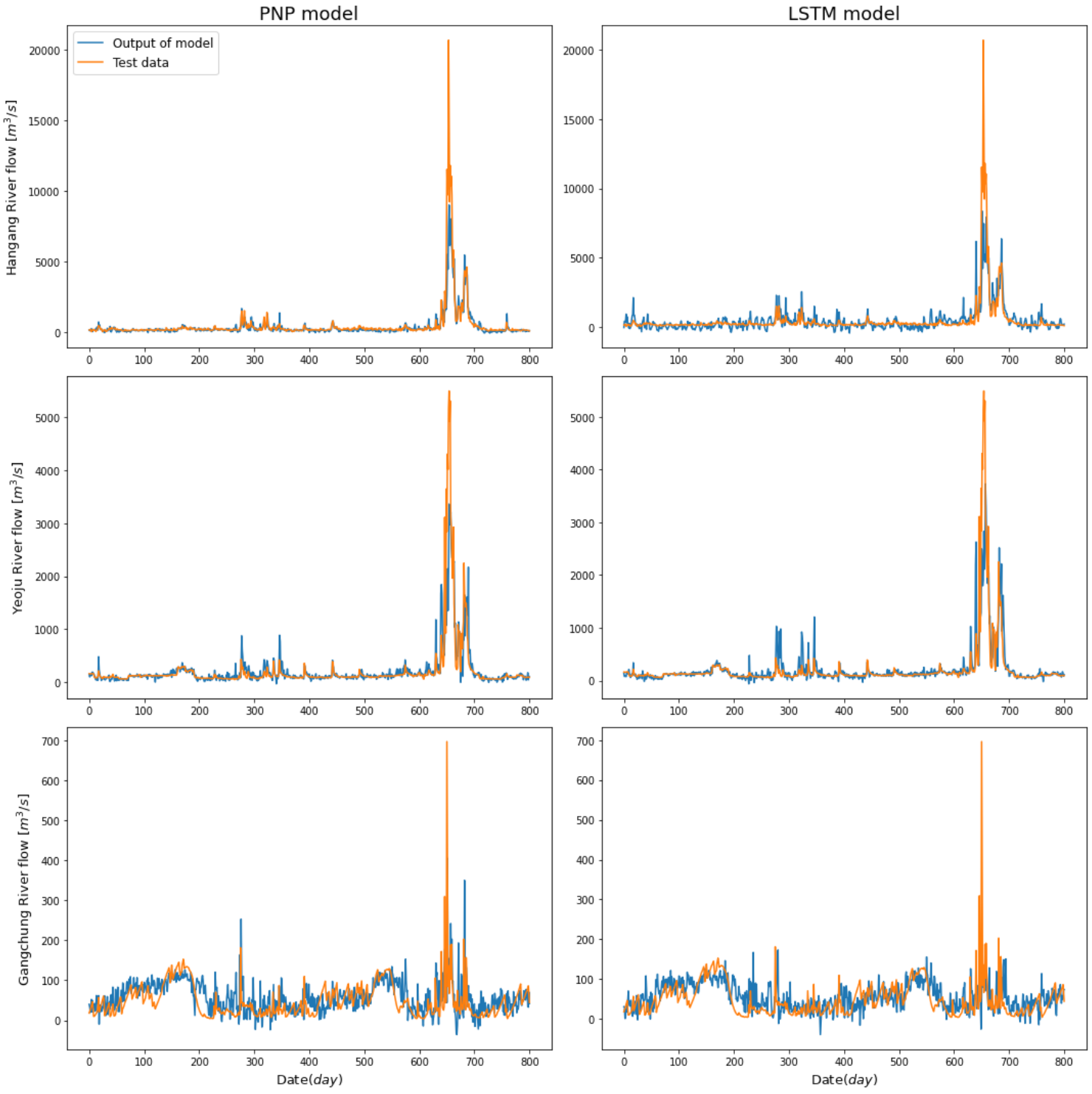


Figure 18

Prediction and observation values of the test data

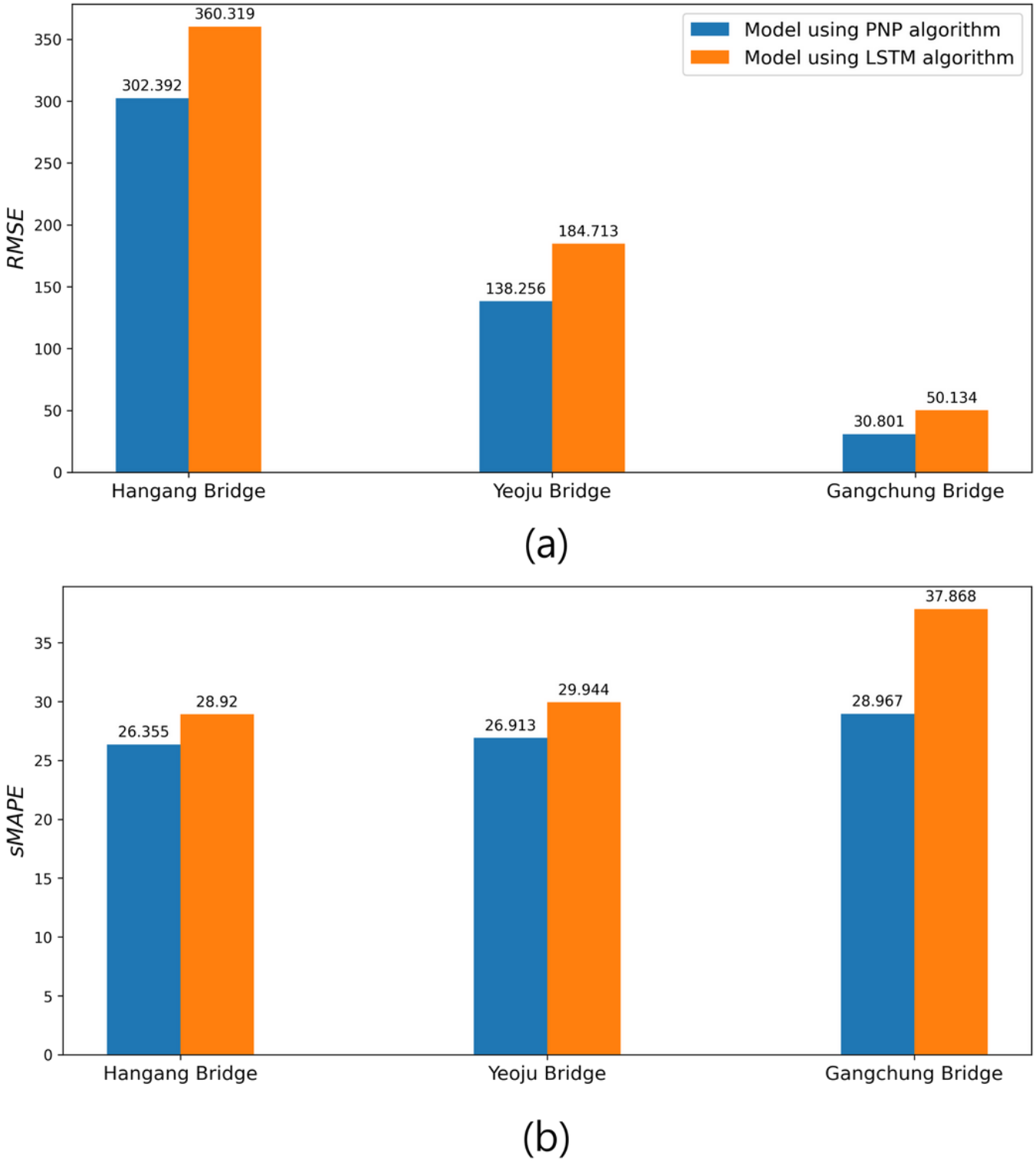


Figure 19

Results of Section 5 models using PNP and LSTM algorithms

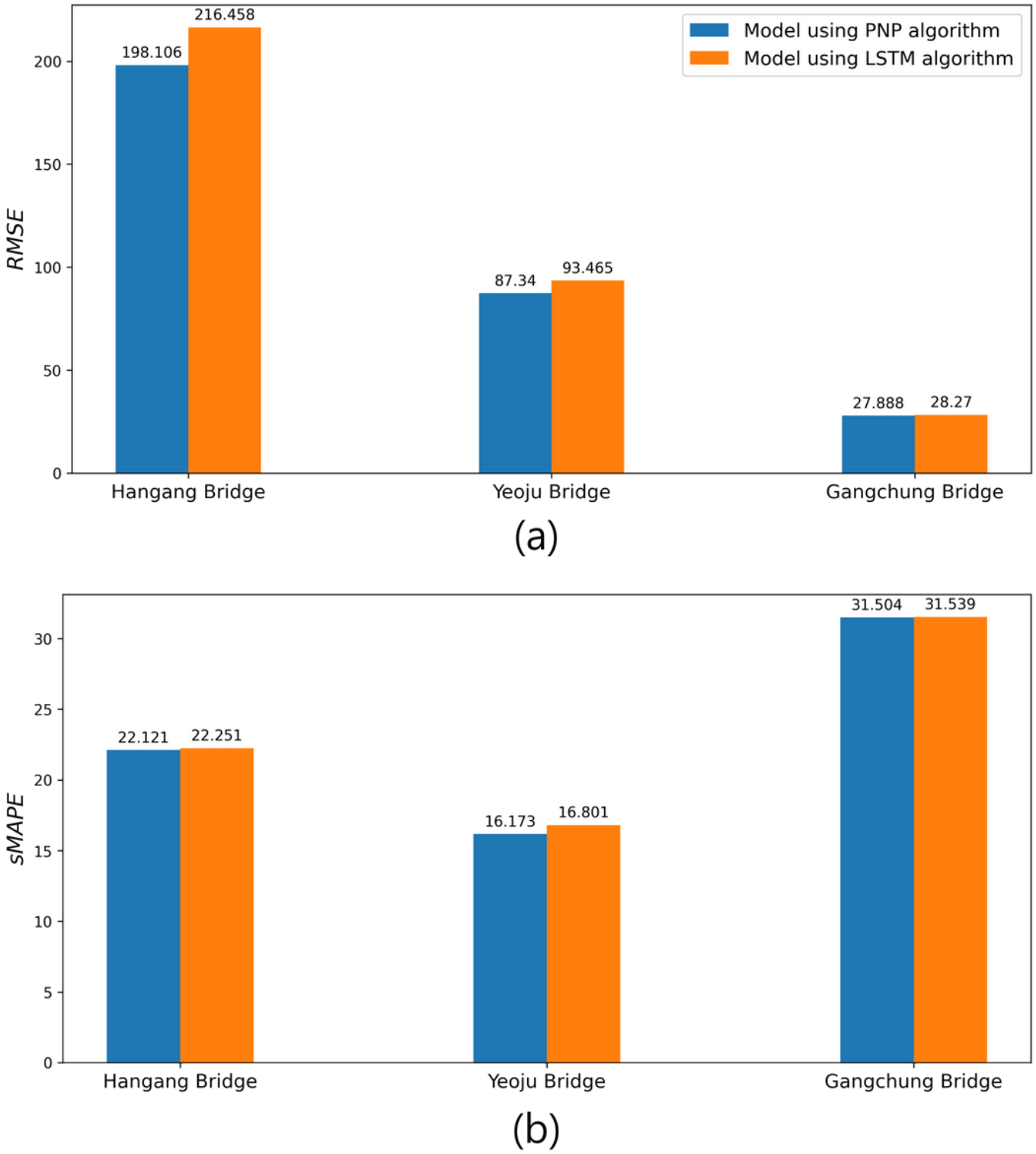


Figure 20

Results of Section 6 models using PNP and LSTM algorithms