Natural Hazards (2023) 116:2339–2361
https://doi.org/10.1007/s11069-022-05766-1

ORIGINAL PAPER

Long short-term memory integrating moving average method for flood inundation depth forecasting based on observed data in urban area

Song-Yue Yang[1] · Bing-Chen Jhong[2] · You-Da Jhong[3] · Tsung-Tang Tsai[4] · Chang-Shian Chen[5]

Received: 5 June 2022 / Accepted: 25 November 2022 / Published online: 21 December 2022

Abstract

Since flooding in urban areas is rarely observed using sensors, most researchers use artificial intelligence (AI) models to predict flood hazards based on model simulation data. However, there is still a gap between simulation and real flooding phenomenon due to the limitation of the model. Few studies have reported on the AI model for flood inundation depth forecasting based on observed data. This study presents a novel method integrating long short-term memory (LSTM) with moving average (MA) for flood inundation depth forecasting based on observed data. A flood-prone intersection in Rende District, Tainan, Taiwan, was adopted as the study area. This investigation compared the forecasting performance of the backpropagation neural network (BPNN), recurrent neural network (RNN) and LSTM models. Accumulated rainfall (R_a) and the moving average (MA) method were applied to enhance the LSTM model performance. The model forecast accuracy was evaluated using root mean square error, coefficient of determination (R^2) and Nash–Sutcliffe efficiency (NSE). Analytical results indicated that the LSTM had better forecasting ability than the RNN and BPNN, because LSTM had both long-term and short-term memory. Since R_a was an important factor in flooding, adding the R_a to the model input upgraded the LSTM forecasting accuracy for high inundation depths. Because MA reduced the noise of the data, processing the model output using the MA also elevated the forecasting accuracy for high inundation depths. For 3-step-ahead forecasting, the NSE of the model benchmark BPNN was 0.79. Using LSTM, R_a and MA, NSEs gradually increased to 0.83, 0.88 and 0.91, respectively.

Keywords Flood inundation depth · Flood forecasting · Deep learning · Long short-term memory · Moving average · Accumulated rainfall · Urban area

✉ You-Da Jhong
ydjhong@fcu.edu.tw; t345869@gmail.com

Extended author information available on the last page of the article

1 Introduction

Between 1998 and 2017, 3148 floods occurred around the world, affecting more than 2 billion people and causing economic losses of US $656 billion (Wallemacq and House 2018). Urbanization has led to dense population and intense economic activity in urban centers, exposing cities to greater risks of flooding. Global climate change has caused sea levels to rise and increased rainstorm intensity in recent years, seriously threatening many major cities around the world (Dieperink et al. 2016; Huong and Pathirana 2013; Ward et al. 2011). Huong and Pathirana (2013) proposed that increased rainfall and rising sea levels due to global climate change cause the water level in the Mekong River to rise. The current degree of urban expansion in Can Tho city, Vietnam, would significantly increase the flood risk. Kefi et al. (2020) estimated future direct flood damage in two urban watersheds, the Ciliwung River in Jakarta, Indonesia, and the Pasig–Marikina–San Juan River in Metro Manila, Philippines. Their analytical results reveal that the total flood damage would have risen by 80% and 212% over current levels in the study areas of Jakarta and Manila, respectively, in 2030. This increase is due to the occurrence of extreme rainfall events and the increase in the degree of urbanization. Flood inundation depth and flooded area significantly affect flood damage.

Accurate flood prediction and forecasting enable the authorities and residents to respond early to reduce the losses and damage caused by floods. Physical models and data-driven methods are often adopted to forecast floods. Physical modeling methods generally involve a rainfall–runoff model to forecast runoff, a one-dimensional hydraulic model for flood routing and a two-dimensional hydraulic model for flood inundation simulation (Chang et al. 2018). Data-driven methods usually require less computing resource than physical models and can be applied for real-time operational flood forecasting. Backpropagation neural network (BPNN) is one of the most frequently used models. BPNN has many advantages, including the ability to process sequence data, high forecasting accuracy and fast recall, and it can handle complex sample identification and nonlinear function synthesis problems. Some studies have found that BPNN model has a good ability to forecast river discharges and water levels (Campolo et al. 1999; Dawson and Wilby 1998; Elsafi 2014; Siou et al. 2011), and sometimes generates even better forecasting results than the rainfall–runoff model (Kerh and Lee 2006).

The main disadvantage of BPNN is that it loses information about the input sequence. The difference between BPNN and recurrent neural network (RNN) is that RNN has at least one feedback loop, and its output is passed back to other neurons in the same layer or the previous layer as input data. Therefore, RNN is suitable for handling static or dynamic time series data. Some studies have utilized RNN to forecast river discharges and water levels (Chang et al. 2002, 2004, 2012; Deshmukh and Ghatol 2010), and show that RNN models have more accurate forecasting results than BPNN model (Chen et al. 2013; Razavi and Karamouz 2007).

The main disadvantage of RNN is that it cannot learn long-term dependencies or remember information for a long time (Guo 2013). Long short-term memory (LSTM) is a special RNN originally developed by Hochreiter and Schmidhuber (1997). The special network architecture of LSTM enables it to learn both short-term and long-term dependencies. LSTM has been widely employed in river discharges and water levels forecasting in recent years (Kratzert et al. 2018). LSTM models have been utilized to forecast the discharges and water levels in different types of rivers, and show excellent forecasting capabilities (Hu et al. 2018; Le et al. 2019; Liu et al. 2020; Song et al. 2020;

Zhang et al. 2018a). The development of complex network architectures can enhance the predictive capabilities of LSTM (Ding et al. 2019; Kao et al. 2020; Kratzert et al. 2019a, b; Ni et al. 2020; Xiang et al. 2020). Heuristic iterative search technology can be used to optimize the parameters of LSTM model to improve the forecasting accuracy (Yuan et al. 2018). Preprocessing input data can improve the forecasting accuracy of LSTM (Ni et al. 2020; Zhu et al. 2020; Zuo et al. 2020). Smoothing time series data using a moving average (MA) algorithm can enhance ANN-based model forecasting. Nhita et al. (2015) used MA to smooth rainfall time series data, increasing the rainfall forecasting accuracy with evolutionary neural network (ENN). Xiang et al. (2020) used MA to process rainfall data to increase the runoff forecasting accuracy by LSTM-based sequence-to-sequence model.

Artificial intelligence (AI) models have been widely adopted in urban flood inundation management (Sayers et al. 2014). Some scholars employed AI models to assess urban flood risk based on geographic information system (GIS) data. Darabi et al. (2019) used genetic algorithm rule-set production (GARP) and quick unbiased efficient statistical tree (QUEST) for flood risk mapping. Rahmati et al. (2019) employed SOM for urban flood hazard mapping. Zhao et al. (2019) utilized weakly labeled support vector machine (WELLSVM) to assess urban flood susceptibility. Lei et al. (2021) adopted convolutional neural network (NNETC) and recurrent neural network (NNETR) for flood hazard mapping.

Some scholars have used simulated data from the flooding simulation model to predict the flood hazard maps, because observed time series data of flood inundation depths are not as easy to obtain. Chang et al. (2010) combined K-means clustering analysis, linear regression model and BPNN to predict the flood hazard maps. Shen and Chang (2013) presented a recurrent configuration of nonlinear autoregressive with exogenous inputs network (R-NARX) to predict the flood inundation depth. Chang et al. (2014) employed self-organizing map (SOM) and R-NARX to obtain the real-time inundation map. Kao et al. (2021) proposed the SAE-RNN model, which combines a stacked autoencoder (SAE) and a RNN to predict the flooding area. Those studies were based on the simulated two-dimensional (2D) data obtained with the model integrated HEC-1, SWMM and 2D non-inertial overland flow simulation models. Jhong et al. (2016) combined support vector machine (SVM) with multi-objective genetic algorithm (MOGA) to predict the flood hazard maps. Jhong et al. (2017) developed an integrated two-stage support vector machine approach to produce flood hazard maps based on simulated 2D data with FLO-2D. However, because of the limitation of the model, there is still a gap between simulation and actual flooding phenomenon. Chang et al. (2018) pointed out that the coupled one-dimensional and two-dimensional hydrodynamics models (1D–2D model) have difficulty in accurately simulating the small-scale flooding with poor road drainage system. The flooding can only be detected by flooding sensors. Since few flooding sensors are installed in cities, time series data of real urban floods are rarely observed and used in AI models for forecasting.

This study has developed a novel LSTM integrating MA for high-temporal-resolution flood inundation depth forecasting based on 10-min-resolution observed data in an urban area. The study was performed at a flood-prone intersection in Rende District, Tainan City, Taiwan. The BPNN, RNN and LSTM models were run to forecast flooding depth, and their forecasting accuracies were compared. Furthermore, two methods were proposed to improve model performance, namely accumulated rainfall (R_a) to improve the input (Jhong et al. 2017) and MA (Nhita et al. 2015; Xiang et al. 2020) to improve the output. Four issues were explored: (1) comparison of the LSTM-, RNN- and BPNN-based models in forecasting accuracy; (2) the addition of accumulated rainfall factor into the LSTM-based model for improving model accuracy; (3) the application of the

MA to process the model output to further improve model accuracy; and (4) using BPNN as the benchmark to compare the improvement effects of LSTM, R_a and MA.

2 Methodology

2.1 Backpropagation neural networks (BPNN)

The architecture of BPNN is mostly multilayer perceptron (MLP), as shown in Fig. 1. The structure can have more than one hidden layer between the input layer and the output layer. The nonlinear mapping relationship between the input layer and the output layer is processed with error backpropagation (EBP). In each neuron, the weight (w_i) sum (*net*) is calculated using its inputs (Xi) and the bias (b), and is then input into a predetermined activation function (σ), as shown below:

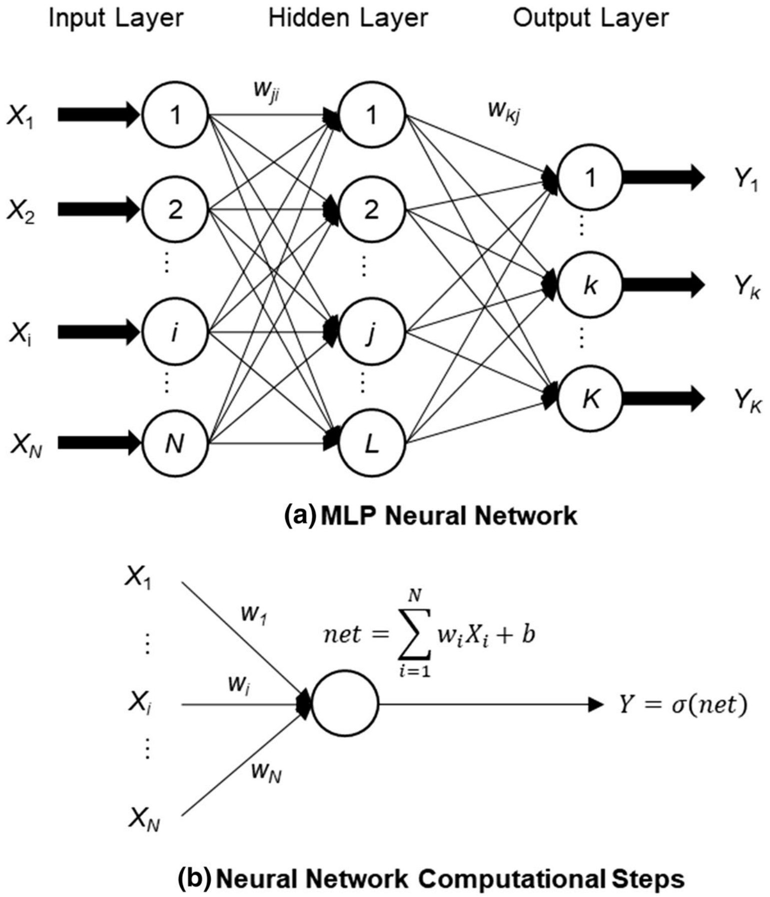


Fig. 1 Architecture of BPNN

$$net = \sum_{i=1}^{N} w_i X_i + b \tag{1}$$

$$Y = \sigma(net) = \max(0, net) \tag{2}$$

The output of one neuron Y is used as the input of neurons in higher layers. In this investigation, ReLu was selected as the activation function. Backpropagation method, based on approximating to a gradient descent method, was applied to adjust the connection weights in the network during the training process.

2.2 Recurrent neural network (RNN)

The architecture of RNN belongs to a three-layer network with an extra set of context unit, as illustrated in Fig. 2a. The context unit is fed back to the input layer from the output vector of the hidden layer. Therefore, besides the original input vector, the input layer also contains the information of the previous hidden layer, enabling the dependency relationship of the time series to be established. This study employed an Elman network, shown as follows:

$$h_t = \sigma_h\left(W_h x_t + U_h h_{t-1} + b_h\right) \tag{3}$$

$$y_t = \sigma_y\left(V_y h_t + b_y\right) \tag{4}$$

where x_t denotes the input vector; h_t represents the hidden layer vector; y_t indicates the output vector; W, U and V are the weight vectors; b denotes the bias vector; and σ_h and σ_y are the activation functions.

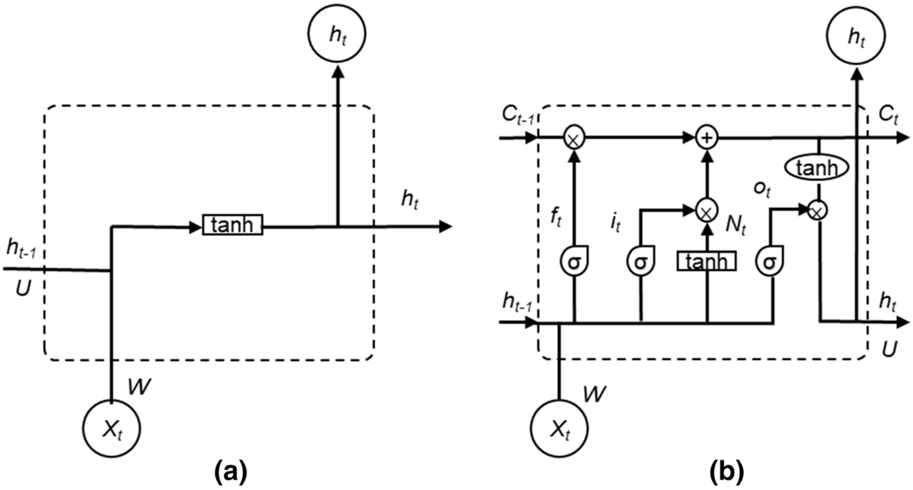


Fig. 2 **a** Architecture of RNN. **b** Architecture of LSTM

2.3 Long short-term memory (LSTM)

LSTM has four interacting layers with a unique communication method. Figure 2b illustrates the structure of LSTM. The LSTM network comprises memory blocks (cells). Two states, namely cell and hidden states, are transferred to the next cell. Data are transferred forward in the cell state. Through the different gates, data can be added or removed in the cell state. The first step is to delete unnecessary information from the cell through the forget gate (f_t), which is a vector corresponding to the cell state C_{t-1}.

$$f_t = \sigma(W_f\left[h_{t-1}, X_t\right] + b_f) \tag{5}$$

where h_{t-1} represents the last unit at time $t-1$; X_t indicates the current input at time t; σ is the activation function. W_f denotes the weight vector of the forget gate; and b_f represents the bias vector of the forget gate. The next step is to compute and store information from the new input (X_t) in the cell state and update the cell state. First, the active function determines whether to update or ignore the new information. Second, the *tanh* function assigns the weight to the passed value to derive its importance level (-1 to $+1$). The new unit state is updated by multiplying the two values. The new state C_t is obtained as the new unit state plus the result of multiplying f_t and C_{t-1}.

$$i_t = \sigma(W_i\left[h_{t-1}, X_t\right] + b_i) \tag{6}$$

$$N_t = tanh(W_n\left[h_{t-1}, X_t\right] + b_n) \tag{7}$$

$$C_t = C_{t-1}f_t + N_t i_t \tag{8}$$

where i_t is input gate; W_i and W_n indicate the weight vectors; bi and b_n are the bias vectors; C_{t-1} is the cell state at time $t-1$; and C_t denotes the cell state at time t. Finally, the output gate (O_t) determines the parts of the unit state to be output. The values can be computed from the new output (h_t), which can be obtained from the O_t multiplied by the value from substituting C_t into the *tanh* function.

$$O_t = \sigma(W_o\left[h_{t-1}, X_t\right] + b_O) \tag{9}$$

$$h_t = O_t tanh\left(C_t\right) \tag{10}$$

where W_o and b_o represent the weight and bias vectors of the output gate, respectively.

2.4 Moving average method

Figure 3 shows the process of the MA method. The steps are listed as follows.

2.4.1 Data preprocessing

Before model forecasting, the MA can be used to process the model output data. The formula of observed average inundation depth ($\overline{D}_{t+\Delta t}$) at time $t+\Delta t$ and $\overline{D}_{t+\Delta t-1}$ at time $t+\Delta t-1$ is shown as:

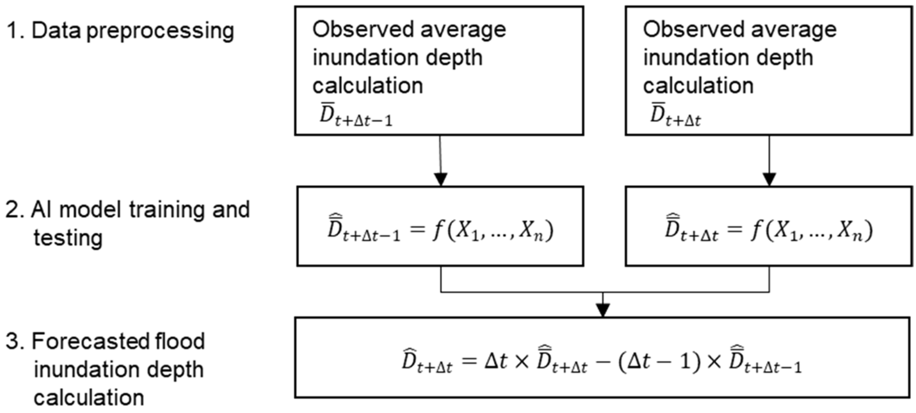

1. Data preprocessing
Observed average inundation depth calculation $\overline{D}_{t+\Delta t-1}$
Observed average inundation depth calculation $\overline{D}_{t+\Delta t}$
2. AI model training and testing
$\widehat{\overline{D}}_{t+\Delta t-1} = f(X_1, \ldots, X_n)$
$\widehat{\overline{D}}_{t+\Delta t} = f(X_1, \ldots, X_n)$
3. Forecasted flood inundation depth calculation
$\widehat{D}_{t+\Delta t} = \Delta t \times \widehat{\overline{D}}_{t+\Delta t} - (\Delta t - 1) \times \widehat{\overline{D}}_{t+\Delta t-1}$

Fig. 3 Process for moving average method

$$\overline{D}_{t+\Delta t} = \frac{1}{\Delta t} \sum_{n=1}^{\Delta t} D_{t+n} \tag{11}$$

$$\overline{D}_{t+\Delta t-1} = \frac{1}{\Delta t - 1} \sum_{n=1}^{\Delta t-1} D_{t+n} \tag{12}$$

where Δt denotes lead time.

2.4.2 AI model training and testing

$\overline{D}_{t+\Delta t}$ And $\overline{D}_{t+\Delta t-1}$ are, respectively, used as the output of two AI models.$X_1, \ldots, X_n$ are the input of AI models. Input and output data are used for AI model training and testing. Forecast average inundation depth $\overline{D}_{t+\Delta t}$ at time $t+\Delta t$ and $\overline{D}_{t+\Delta t-1}$ at time $t+\Delta t-1$ are shown as:

$$\widehat{\overline{D}}_{t+\Delta t} = f(X_1, \ldots, X_n) \tag{13}$$

$$\widehat{\overline{D}}_{t+\Delta t-1} = f(X_1, \ldots, X_n) \tag{14}$$

where f represents the AI model and $X_1, \ldots, X_n$ indicate model inputs.

2.4.3 Forecast flood inundation depth calculation

After the model forecast is completed, the average flood inundation depths forecasted by the model are converted to the forecasted flood inundation depth. The formula of the forecasted flood inundation depth ($\widehat{D}_{t+\Delta t}$) at time $t+\Delta t$ is as follows:

$$\widehat{D}_{t+\Delta t} = \Delta t \times \widehat{\overline{D}}_{t+\Delta t} - (\Delta t - 1) \times \widehat{\overline{D}}_{t+\Delta t-1} \tag{15}$$

3 Materials

3.1 Case study

The flood inundation depth gauge (FIDG) was set up at a flood-prone intersection in Rende District, Tainan, Taiwan, as shown in Fig. 4. Due to storms from monsoons and typhoons, the intersection is often flooded, causing traffic disruptions. The intersection is 200 m west of the Jitan Bridge. The Jitan Bridge is at the confluence of the Dawan Drainage and Taizimiao Drainage. The drainage at the Jitan Bridge has width of about 20 m and height of about 5 m. The catchment area of the Jitan Bridge is approximately 1,521 hectares. The length of the Dawan Drainage is about 1871 m, and the slope is about 1/850. The length of the Taizimiao Drainage is about 2884 m, and the slope is about 1/740. Several flooding incidents have recently occurred at this site. The heavy rainfall raises the stage of the Dawan Drainage, preventing the runoff in connected sewers from draining into the Dawan Drainage, leading eventually to waterlogging.

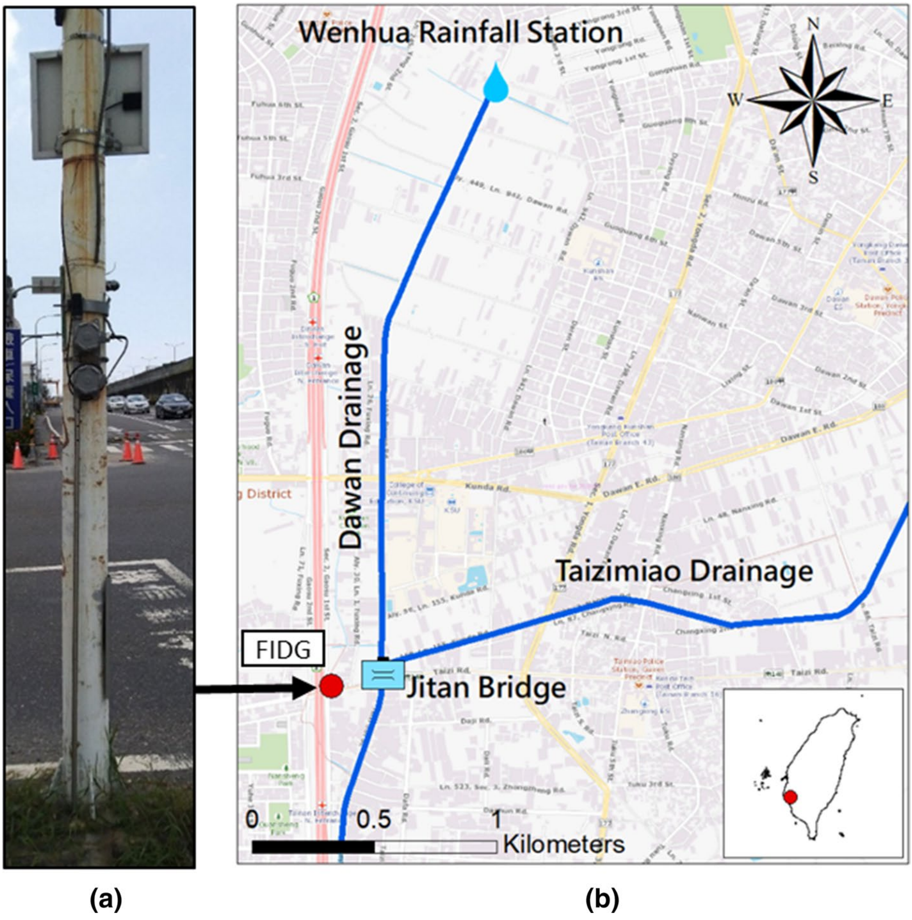


Fig. 4 **a** Flood inundation depth gauge (FIDG). **b** Location of case study in Rende District, Tainan, Taiwan

3.2 Data used

The rainfall observation data with ten-minute interval come from the Wenhua Rainfall Station. The FIDG is an Anasystem SenSmart WLS and observes the flood inundation depth by radio frequency admittance. The accuracy of the FIDG is 0.5% of the length of the sensor (usually 1.5 to 2.0 m). The observed data were transmitted to the cloud server through Long Range (LoRa) every 30 s. The road adjacent to the FIDG erected location is often closed due to flooding. The FIDG was erected in 2017 to record the flooding process and is still under observation. Flooding events caused by one typhoon and four storms were observed, as listed in Table 1 and Fig. 5. Since Event 3 had the longest flooding duration and the second highest maximum flood inundation depth, it was selected as the test event. The other four events were training events. The 575 observations were divided into the training data (328 from 4 events) and the testing data (247 from 1 event). The resolution of the observations was 10 min. The training data set was utilized to adjust the parameters of the models. The test data set was adopted to measure the performance of the models.

3.3 Model benchmark

To highlight the improvement due to the use of LSTM, this study employed the traditional BPNN-based model as the model benchmark. Figure 5 shows a high correlation between rainfall and flood inundation depth. Therefore, this study took observed rainfalls and flood inundation depth as model inputs. The general form is shown as follows:

$$\widehat{D}_{t+\Delta t} = f_{BPNN}(R_t, R_{t-1}, \dots, R_{t-(L_R-1)}, D_t) \tag{16}$$

where R_t is observed rainfall at time t; $R_{t\text{-}1}$ is observed rainfall at time t-1; $R_{t-(L_R-1)}$ is observed rainfall at time t-(L_R-1); L_R represents the lag length of rainfall; and D_t is observed flood inundation depth at time t.

3.4 Model development

Three experiments were designed to evaluate improvement for different methods of the model forecasting ability. The purpose of the first experiment was to evaluate the forecasting capabilities of Model RNN and LSTM, and the general form is shown as:

$$\widehat{D}_{t+\Delta t} = f_{RNN}(R_t, R_{t-1}, \dots, R_{t-(L_R-1)}, D_t) \tag{17}$$

$$\widehat{D}_{t+\Delta t} = f_{LSTM}(R_t, R_{t-1}, \dots, R_{t-(L_R-1)}, D_t) \tag{18}$$

The second experiment was to evaluate improvement of the model forecasting ability by using the R_a as the input. R_a can significantly improve the accuracy of AI model for forecasting flood inundation depth (Jhong et al. 2017). The formula is as follows:

$$R_a = \sum_{n=0}^{L_R-1} R_{t-n} \tag{19}$$

The general form is shown as:

Table 1 Summary statistics of the observed typhoon and storms

Model stage	ID	Event	Time	Maximum 60-min accumulated rainfall (mm)	Accumulated rainfall (mm)	Rainfall duration (hr)	Peak inundation depth (mm)	Inundation duration (hr)	Number of each data set
Training	1	Storm 0619	2018/6/19	41	331	22.2	49	0.7	137
	2	Storm 0620	2018/6/20	29	126	10.5	80	1.0	64
	4	Storm 0611	2019/6/11	65	121	6.8	156	1.0	42
	5	Storm 0813	2019/8/13	98	297	14.0	574	9.7	85
Testing	3	Typhoon Mangkhut	2018/8/23	65	601	51.0	374	23.7	247

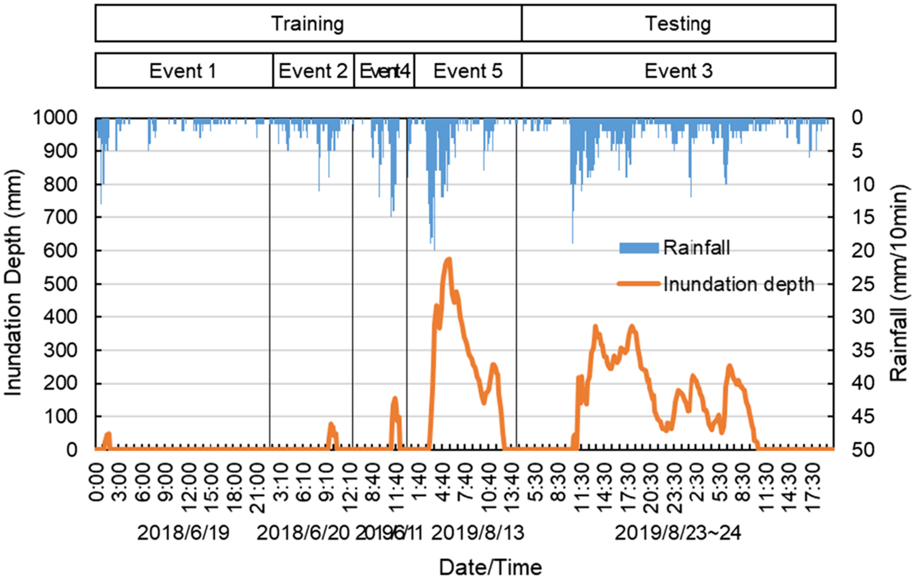


Fig. 5 Observed inundation events for model training and testing

$$\hat{D}_{t+\Delta t} = f_{LSTM}(R_a, D_t) \tag{20}$$

The third experiment was to evaluate the improvement of the model forecasting ability by using MA. The general forms are shown as Eq. (21) and (22).

$$\hat{\bar{D}}_{t+\Delta t} = f_{LSTM}(R_a, D_t) \tag{21}$$

$$\hat{\bar{D}}_{t+\Delta t-1} = f_{LSTM}(R_a, D_t) \tag{22}$$

After the model forecast was completed, the forecasted flood inundation depths were converted according to Eq. (15).

The correlation coefficients between the flood inundation depth and rainfalls with different L_R were calculated to find the appropriate L_R. Then, the trial-and-error method was further used to find the optimal L_R. Therefore, L_R was selected as 6 in this study. Δt was selected as 3 in this study. Table 2 lists the names, algorithms, inputs and outputs of models.

The BPNN and LSTM models were programmed in Python 3.8 with the Keras database. The data were normalized with the max–min scaler. The program was compiled using the Adam optimizing compiler. The optimal hyper-parameters, such as numbers of hidden layers and nodes, of the model were obtained by trial and error. All models had 1 hidden layer and 20 neurons. We solved the local problem by running multiple simulations. Take $t+3$ forecasting as an example, the convergence plots of BPNN, RNN, LSTM are shown in Fig. 6. The computation times of BPNN, RNN, LSTM were 2.23, 6.43 and 11.45 s, respectively.

Table 2 Names, algorithms, inputs and outputs of models

Name	Algorithm	Input	Output
BPNN	BPNN	$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+1}$
		$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+2}$
		$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+3}$
RNN	RNN	$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+1}$
		$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+2}$
		$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+3}$
LSTM	LSTM	$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+1}$
		$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+2}$
		$D_{\mathrm{t}}, R_{\mathrm{t}}, R_{\mathrm{t}-1}, R_{\mathrm{t}-2}, R_{\mathrm{t}-3}, R_{\mathrm{t}-4}, R_{\mathrm{t}-5}$	$\widehat{D}_{t+3}$
LSTM-Ra	LSTM	D_{t}, R_a	$\widehat{D}_{t+1}$†
		D_{t}, R_a	$\widehat{D}_{t+2}$
		D_{t}, R_a	$\widehat{D}_{t+3}$
LSTM-Ra-MA	LSTM	D_{t}, R_a	$\widehat{D}_{t+1}$†
		D_{t}, R_a	$\widehat{\overline{D}}_{t+2}$
		D_{t}, R_a	$\widehat{\overline{D}}_{t+3}$

D_{t}: observed flood inundation depth at time t; $\widehat{D}_{t+1}$–$\widehat{D}_{t+3}$: forecasted flood inundation depths at time $t+1$–$t+3$; R_{t}: observed flood inundation depth at time t; $R_{\mathrm{t}-1}$–$R_{\mathrm{t}-5}$: observed rainfalls at time t-1 ~ t-5; R_a: accumulated rainfall; $\overline{D}_{t+2}$ and $\overline{D}_{t+3}$: forecasted average inundation depths at $t+2$ and $t+3$

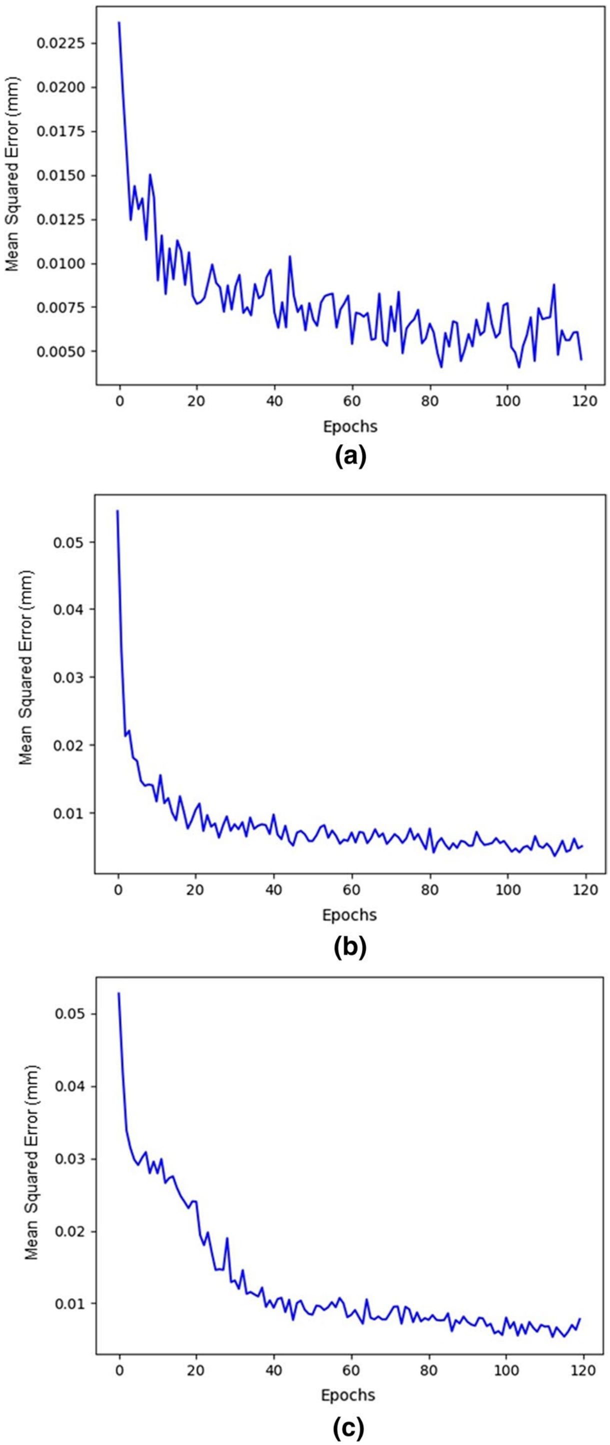
†The predicted results of LSTM-Ra and LSTM-Ra-MA are the same for $\widehat{D}_{t+1}$ forecasting

3.5 Evaluation of model performance

The forecasting performance of models was measured from the root mean square error (RMSE), coefficient of determination (R^2) and Nash–Sutcliffe efficiency (NSE). The RMSE value indicates the error between the forecasted and observed values. An RMSE value closer to 0 indicates a more accurate forecasting. The R^2 value measures the linear correlation between the forecasted and observed values, and ranges from 0 to 1. An R^2 value closer to 1 indicates a more accurate forecasting. The value of NSE is generally utilized to evaluate the forecasting of hydrological models, and ranges from $-\infty$ to 1. An NSE value closer to 1 indicates a more accurate forecasting. The formulae of the above evaluation indices are as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\widehat{D}_i - D_i\right)^2} \tag{23}$$

Fig. 6 **a** Convergence plot of BPNN. **b** Convergence plot of RNN. **c** Convergence plot of LSTM



$$R^2 = \left[\frac{\sum_{t=1}^{N} (\widehat{D}_i - \overline{\widehat{D}_i})(D_i - \overline{D_i})}{\sqrt{\sum_{t=1}^{N} (\widehat{D}_i - \overline{\widehat{D}_i})^2 \sum_{t=1}^{N} (D_i - \overline{D_i})^2}} \right]^2 \tag{24}$$

$$\mathrm{NSE} = 1 - \frac{\sum_{t=1}^{N} (D_i - \widehat{D}_i)^2}{\sum_{t=1}^{N} (D_i - \overline{D_i})^2} \tag{25}$$

$$ED_p = D_p^{for} - D_p^{obs} \tag{26}$$

where $\widehat{D}_i$ and D_i represent, respectively, the ith forecasted and observed flood inundation depths, and $\overline{\widehat{D}_i}$ and $\overline{D_i}$ are, respectively, the means of forecasted and observed flood inundation depths. D_p^{for} and D_p^{obs} are, respectively, the peak forecasted and observed flood inundation depths.

4 Results and discussion

4.1 Improvement due to the use of LSTM model

Table 3 lists the performance of models in the training stage. All R^2 and NSE values were greater than 0.91, indicating that the models were well trained. Figure 7a–c illustrates the comparison of different indicators between the BPNN, RNN and LSTM in the test stage. Those results revealed that the accuracy of the three models decreased as the lead time increased. For $t+1$ and $t+2$ forecastings, RNN outperformed LSTM, which outperformed BPNN. However, for $t+3$ forecasting, LSTM was better than BPNN and than RNN.

Figure 7d shows that BPNN performs very close to RNN and LSTM in peak forecasting; however, the BPNN obviously underestimated observed values below 100 mm. Figure 5 shows the rising and falling limbs of flood inundation depth. When the rain reached a certain level, the flood inundation depth increased rapidly with the rainfall. When the rain stopped, the depth decreased slowly. The flooding on road was discharged into the side ditch first and then discharged into the storm sewer through the connecting pipes. This is a nonlinear system. BPNN does not have a memory function, resulting in inability to forecast this nonlinear system. LSTM and RNN can predict this nonlinear system because of their memory function. Hu et al. (2018) used BPNN and LSTM to simulate the rainfall–runoff process of Fen River in China. This study indicated that when the simulated flow was low, the simulated values of BPNN model fluctuate abnormally. The simulated values of LSTM model were more stable than those of BPNN model. Therefore, the LSTM model has better nonlinear simulation ability. Zhang et al. (2018b) observed that the performance of BPNN degraded as the lead time increased, and BPNN, a static feed-forward ANN, was not suitable for multiple-step-ahead forecasting. These findings are consistent with our experimental results.

For $t+1$ and $t+2$ forecasting, RNN performed slightly better than LSTM, but for $t+3$ forecasting, LSTM performed better than RNN. Because of the vanishing or exploding

Table 3 Performance of models

Model	Lead time	Training			Testing		
		RMSE (mm)	R^2	NSE	RMSE (mm)	R^2	NSE
BPNN	1	0.04	0.99	0.97	35.46	0.96	0.91
	2	0.04	0.98	0.97	41.22	0.91	0.87
	3	0.04	0.97	0.97	52.65	0.86	0.79
RNN	1	0.03	0.99	0.99	23.92	0.97	0.96
	2	0.04	0.98	0.97	36.69	0.93	0.90
	3	0.04	0.97	0.97	53.68	0.88	0.79
LSTM	1	0.03	0.99	0.98	30.81	0.97	0.93
	2	0.03	0.98	0.98	40.01	0.93	0.88
	3	0.04	0.97	0.96	48.01	0.87	0.83
LSTM-Ra	1†	0.03	0.98	0.98	19.90	0.97	0.97
	2	0.05	0.95	0.95	32.94	0.94	0.92
	3	0.07	0.91	0.91	40.77	0.89	0.88
LSTM-Ra-MA	1†	0.03	0.98	0.98	19.90	0.97	0.97
	2	0.04	0.97	0.97	27.52	0.96	0.94
	3	0.05	0.96	0.96	33.87	0.93	0.91

†The predicted results of LSTM-Ra and LSTM-Ra-MA are the same for $\widehat{D}_{t+1}$ forecasting

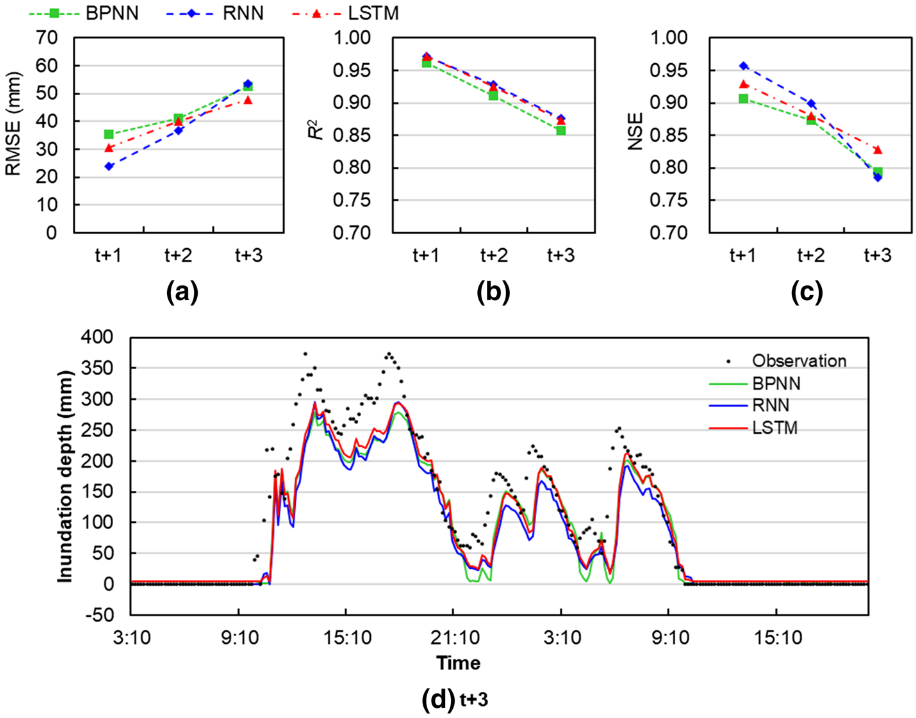


Fig. 7 **a–c** Performance of BPNN, RNN and LSTM in testing stage. **d** Comparisons of observed and forecasted inundation depth from LSTM, RNN and LSTM at 3 steps ahead in testing stage

gradients, RNNs can only learn dependencies for 10 or fewer time steps (Bengio et al. 1994; Hochreiter and Schmidhuber 1997). Because this study only forecasted 3 steps ahead, the forecasting accuracies of RNN and LSTM were similar. When the forecast lead time is longer, LSTM should perform better than RNN. Kratzert et al. (2018) used LSTM to predict runoffs in 241 watersheds in USA. Their study found that RNN had good forecasting with short lead time of forecasting, but LSTM outperformed RNN with long lead time. Their work made similar inferences.

4.2 Improvement due to the use of the LSTM-based model with the accumulated rainfall

Figure 8a–c compares different indicators running LSTM and LSTM-Ra in the test stage. The RMSE values of the LSTM were, respectively, 31 mm, 40 mm and 48 mm, and the RMSE values of the LSTM-Ra were, respectively, 20 mm, 33 mm and 41 mm, for 10-, 20- and 30-min-ahead forecasting. Adopting R_a as the model input reduced RSME values for LSTM by 35.4%, 17.7% and 15.1%, respectively. All R^2 and NSE values of the LSTM-Ra were larger than those the LSTM, respectively. These results showed that using R_a as input could significantly upgrade the accuracy of LSTM forecasting.

Figure 8d shows that the forecasted values of LSTM-Ra are closer than those of LSTM to the observed values, especially at high values. For 10-, 20- and 30-min-ahead forecasting, the ED_p, the maximum forecasted depth minus the maximum observed depth, of

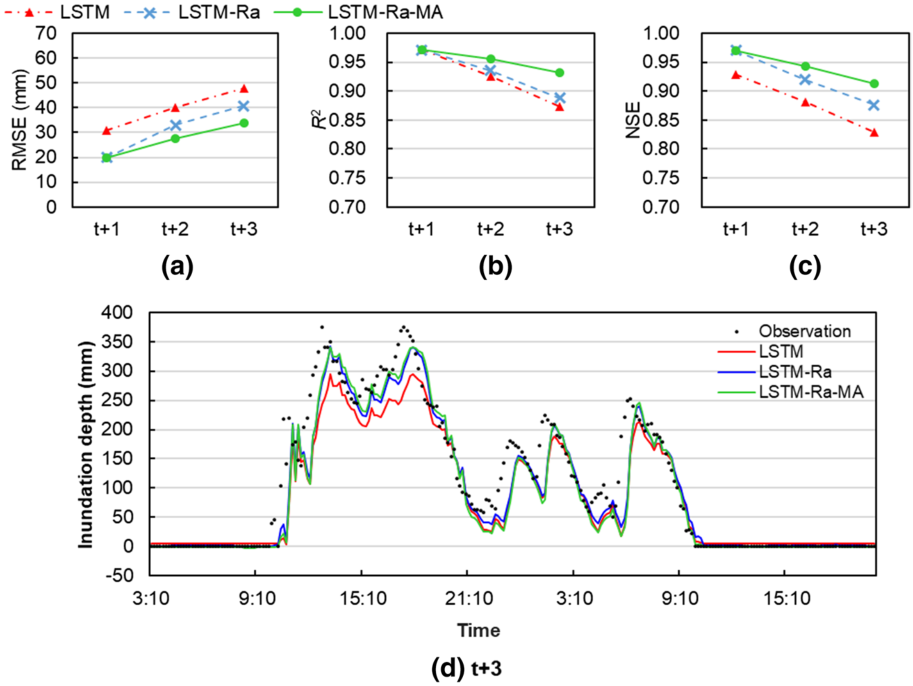


Fig. 8 **a**–**c** Performance of LSTM, LSTM-Ra and LSTM-Ra-MA in testing stage. **d** Comparisons of observed and forecasted inundation depth from LSTM, LSTM-Ra and LSTM-Ra-MA at 3 steps ahead in testing stage

the LSTM-Ra were − 18 mm, − 37 mm and − 32 mm, respectively, and their biases were smaller than − 71 mm, − 89 mm and − 79 mm of the LSTM. This result demonstrated that the accumulated rainfall factor could effectively improve the LSTM-based model's ability to forecast high depths.

Events 1, 2 and 4 in Table 1 and Fig. 5 indicate that the flood inundation depth is based on the amount of rainfall in a short period of time, and is zero if the amount of rainfall is below a bound. The waterlogging only occurred when the storm sewer was filled with floodwater and surface runoff could no longer drain into the storm sewer. Therefore, waterlogging did not occur if the rainfall intensity was not large enough to exceed the flood conveyance capacity of the storm sewer. Event 5 was a typical short-duration intense rainfall, with a rainfall intensity of 98 mm over 60 min. Event 3 was a typical typhoon rainfall. The total rainfall was 601 mm, and the rainfall duration was 51 h. Although Event 5 (9.7 h) had a shorter inundation time than Event 3 (23.7 h), it had a greater inundation depth (574 mm) than Event 3 (374 mm). Jhong et al. (2017) utilized a support vector machine (SVM) to forecast the depth and extent of flooding during the next 1–6 h based on simulated data from a 2D numerical model. Their study proposed that R_a could significantly improve forecast accuracy. The above views are consistent with the findings of our investigation.

4.3 Improvement due to the use of the LSTM-based model with MA

Figure 8a–c illustrates the comparison of different indicators between the LSTM-Ra and the LSTM-Ra-MA in the test stage. The forecastings of the LSTM-Ra and LSTM-Ra-MA were the same for $\widehat{D}_{t+1}$ forecasting. For $\widehat{D}_{t+2}$ and $\widehat{D}_{t+3}$ forecasting, the RMSE values of the LSTM-Ra were 33 mm and 41 mm, respectively, and the RMSE values of the LSTM-Ra-MA were 28 mm and 34 mm, respectively. The MA reduced RMSE values by 16.5% and 16.9% of the LSTM-Ra, respectively. For 20- and 30-min-ahead forecasting, all R^2 and NSE values of the LSTM-Ra-MA were greater than the LSTM-Ra, respectively. Such results showed that using MA to process the flooding depth data could effectively improve the overall forecasting accuracy of the LSTM-based model.

Figure 8d shows that the forecasted value of LSTM-Ra-MA is closer to the observed value than that of LSTM-Ra at high values. For $\widehat{D}_{t+2}$ and $\widehat{D}_{t+3}$ forecasting, the ED_p of LSTM-Ra-MA were, respectively, − 24 mm and − 34 mm for 20- and 30-min-ahead forecasts, and their bias were slightly smaller than the − 37 mm and − 32 mm of LSTM-Ra. This result demonstrated that MA improves the forecasting ability of LTSM-based model for high depth.

Because of the high resolution (10 min) of the observational data used in this study, the observational data had high noise. The model output was processed using MA, and MA could reduce the noise of the data and highlight the data trend. Nhita et al. (2015) used MA to smooth the rainfall time series data, which could improve the forecasting effect of ANN-based model. Xiang et al. (2020) pointed out that using MA to process hourly rainfall data could avoid high variances and noisy signals generated by storms. These observations are consistent with our findings.

4.4 Improvement comparison among LSTM-based model, R_a and MA

Figure 9a plots the RMSE for Models BPNN, LSTM, LSTM-Ra and LSTM-Ra-MA. This graph indicates that the model benchmark gradually improved by method in order of LSTM-based model, R_a and MA, with decreasing RMSE.

In the $\widehat{D}_{t+1}$ forecasting, the RMSE value was reduced from 35 to 20 mm in total. Of the total reduction, 29.9% was contributed by the LSTM-based model and 70.1% by R_a (see Fig. 9b). Because the forecastings of the LSTM-Ra and LSTM-Ra-MA were the same for $\widehat{D}_{t+1}$ forecasting, only the LSTM-based model and R_a were compared. The RMSE value was reduced from 41 to 28 mm in total in the $\widehat{D}_{t+2}$ forecasting. Of the total reduction in RMSE value, 8.8% was contributed by the LSTM-based model, 51.6% by R_a and 39.6% by the MA. In the $\widehat{D}_{t+3}$ forecast, the RMSE value was reduced from 53 to 34 mm in total. Of the total reduction in RMSE value, 24.7% was contributed by the LSTM-based model, 38.6% by R_a and 36.7% by MA.

Among the three methods to improve the accuracy of model forecasting, R_a was the most effective, MA was the second, and the LSTM-based model was the third. This result was also consistent with the physical phenomenon of flooding depth, where the accumulation of sufficient rainfall within a certain period of time causes flooding to occur in flooded areas. MA could reduce the noise of flooding data with high temporal resolution, simplifying flooding trend forecasting for the forecasting model. LSTM-based model could forecast the nonlinear system of flood inundation depth because of its memory function. We suggest that in future studies, R_a should be included in the input. Additionally, the output can be processed using MA and LSTM-based model can be applied as the forecasting model to increase the forecasting accuracy.

4.5 Limitations of the work and future research

Only five observational flood events have occurred so far since the station was established in 2017, further revealing the difficulty in obtaining these data. Higher-quality observations can effectively improve the accuracy of deep learning models. Besides continuous observations in the future, other deep learning models could be applied to reproduce more data to improve the accuracy of observations.

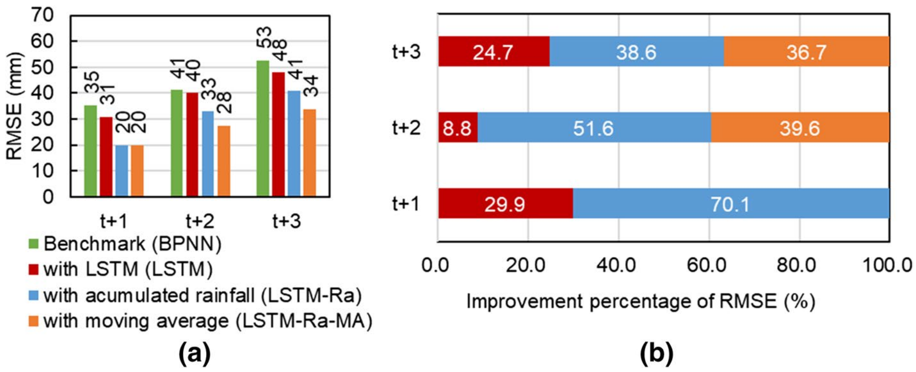


Fig. 9 **a** RMSE of models using different methods in testing stage. **b** RMSE improvement percentage of model prediction using different methods in testing stage

The forecasting lead time of the model in this study is short, leading to limitations in practical application. The accuracy of forecasting drops significantly when running the proposed model with long forecasting lead times. The novel deep learning models could be employed to extend the lead time of forecasting.

In this study, observations from surface rainfall stations were utilized as input to the forecasting model. Model forecasting might be advanced if quantitative rainfall forecast data, such as QPESUM (Quantitative Precipitation Estimation and Segregation Using Multiple Sensors) and QPF (Quantitative Precipitation Forecast), are adopted as model inputs (Chiou et al. 2005; Wang et al. 2016). Urban flooding characteristics are determined by many factors, such as drainage system, vegetation and topography. However, this study only used rainfall and flood inundation depth as model inputs. Future works could incorporate a wider variety of inputs to reflect flooding characteristics.

4.6 impact and usefulness of work concerning water resources management

This model, programmed using Python, could be incorporated into the urban flood control system with an Application Programming Interface (API) in the future. An alarm could be sent to people's mobile phones when the forecast flood inundation depth exceeds the threshold (for example, 50 mm), giving local residents sufficient time to erect waterproof gates at the entrances of their houses. The authority can quickly regulate flooded roads to stop vehicles from entering.

5 Conclusions

This work proposes a novel LSTM integrating MA method for flood inundation depth forecasting based on 10 min-resolution observed data in an urban area. The proposed model accurately forecasted the high-temporal-resolution flooding depth in the flood-prone crossroad in Rende District, Tainan, Taiwan. The research is summarized as follows:

1. The LSTM network with short- and long-term memory was more suitable than the BPNN and RNN networks for forecasting the flood inundation depth.
2. Flood inundation resulted from the accumulated rainfall over a period of time. The model input adopted R_a, which could improve the forecasting ability of the LSTM-based model for high flooding depth.
3. The model output was processed using MA to streamline the flood inundation depth data. The trend of such data could be easily forecasted by models. MA could enhance the forecasting accuracy of LSTM-based model for high inundation depth.
4. Taking BPNN as the benchmark, R_a provided the largest improvement in model forecasting accuracy, followed by MA, and then the LSTM-based model.

Since the observational data of flood inundation depths are quite hard to obtain, this investigation only used 328 data from four rainfall events for model training. The model accuracy can be improved by measuring more actual observations in the future. Other observational data, such as quantitative precipitation forecast and water level data, can be further adopted in future work. The application of other novel deep learning models can also improve the accuracy of the model.

The proposed model can be connected to the flood prevention system of Tainan City. Residents and authorities can follow flood prevention and evacuation measures when forecasted flood inundation depths exceed thresholds. The proposed model can be applied in other domains, such as river flow, water level and groundwater table.

Acknowledgements The authors would like to thank the National Science and Technology Council, Taiwan, for financially supporting this research under Contract No. MOST 109-2625-M-035-007-MY3. The authors thank the Water Resources Planning Institute, Water Resources Agency, for providing relevant information and Mr. George Chih-Yu Chen for English editing.

Author contributions S-YY was involved in conceptualization, data curation, methodology, visualization, writing the original draft, and funding acquisition; B-CJ took part in writing—reviewing and editing; Y-DJ and C-SC contributed to methodology; and T-TT was responsible for software.

Funding This research project is funded by the National Science and Technology Council, Taiwan (Grant numbers MOST 109-2625-M-035-007-MY3).

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

References

Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(157):166

Campolo M, Andreussi P, Soldati A (1999) River flood forecasting with a neural network model. Water Resour Res 35:1191–1197. https://doi.org/10.1029/1998WR900086

Chang FJ, Chang LC, Huang HL (2002) Real-time recurrent learning neural network for stream-flow forecasting. Hydrol Process 16(2577):2588. https://doi.org/10.1002/hyp.1015

Chang LC, Chang FJ, Chiang YM (2004) A two-step-ahead recurrent neural network for stream-flow forecasting. Hydrol Process 18:81–92. https://doi.org/10.1002/hyp.1313

Chang L-C, Shen H-Y, Wang Y-F, Huang J-Y, Lin Y-T (2010) Clustering-based hybrid inundation model for forecasting flood inundation depths. J Hydrol 385(257):268. https://doi.org/10.1016/j.jhydrol.2010.02.028

Chang L-C, Chen P-A, Chang F-J (2012) Reinforced two-step-ahead weight adjustment technique for online training of recurrent neural networks. IEEE Trans Neural Netw Learn Syst 23(1269):1278. https://doi.org/10.1109/TNNLS.2012.2200695

Chang L-C, Shen H-Y, Chang F-J (2014) Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. J Hydrol 519:476–489. https://doi.org/10.1016/j.jhydrol.2014.07.036

Chang C-H, Chung M-K, Yang S-Y, Hsu C-T, Wu S-J (2018) A case study for the application of an operational two-dimensional real-time flooding forecasting system and smart water level gauges on roads in Tainan City, Taiwan. Water 10:574. https://doi.org/10.3390/w10050574

Chen P-A, Chang L-C, Chang F-J (2013) Reinforced recurrent neural networks for multi-step-ahead flood forecasts. J Hydrol 497:71–79. https://doi.org/10.1016/j.jhydrol.2013.05.038

Chiou PT-K, Chen C-R, Chang P-L, Jian G-J (2005) Status and outlook of very short range forecasting system in Central Weather Bureau, Taiwan. In: Applications with Weather Satellites II. International Society for Optics and Photonics, pp 185–197. doi:https://doi.org/10.1117/12.601195

Darabi H, Choubin B, Rahmati O, Haghighi AT, Pradhan B, Kløve B (2019) Urban flood risk mapping using the GARP and QUEST models: a comparative study of machine learning techniques. J Hydrol 569(142):154. https://doi.org/10.1016/j.jhydrol.2018.12.002

Dawson CW, Wilby R (1998) An artificial neural network approach to rainfall-runoff modelling. Hydrol Sci J 43:47–66. https://doi.org/10.1080/02626669809492102

Deshmukh RP, Ghatol A (2010) Short term flood forecasting using recurrent neural networks a comparative study International Journal of. Eng Technol 2:430

Dieperink C, Hegger DT, Bakker M, Kundzewicz ZW, Green C, Driessen P (2016) Recurrent governance challenges in the implementation and alignment of flood risk management strategies: a review. Water Resour Manage 30:4467–4481. https://doi.org/10.1007/s11269-016-1491-7

Ding Y, Zhu Y, Wu Y, Jun F, Cheng Z (2019) Spatio-Temporal Attention LSTM Model for Flood Forecasting. In: 2019 International conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, pp 458–465. doi:https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00095

Elsafi SH (2014) Artificial neural networks (ANNs) for flood forecasting at Dongola Station in the River Nile, Sudan. Alex Eng J 53:655–662. https://doi.org/10.1016/j.aej.2014.06.010

Guo J (2013) Backpropagation through time Unpubl ms. Harbin Institute of Technology 40:1–16

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu C, Wu Q, Li H, Jian S, Li N, Lou Z (2018) Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water 10:1543. https://doi.org/10.3390/w10111543

Huong H, Pathirana A (2013) Urbanization and climate change impacts on future urban flooding in Can Tho city, Vietnam. Hydrol Earth Syst Sci 17:379–394. https://doi.org/10.5194/hess-17-379-2013

Jhong B-C, Wang J-H, Lin G-F (2016) Improving the long lead-time inundation forecasts using effective typhoon characteristics. Water Resour Manage 30:4247–4271. https://doi.org/10.1007/s11269-016-1418-3

Jhong B-C, Wang J-H, Lin G-F (2017) An integrated two-stage support vector machine approach to forecast inundation maps during typhoons. J Hydrol 547:236–252. https://doi.org/10.1016/j.jhydrol.2017.01.057

Kao I-F, Zhou Y, Chang L-C, Chang F-J (2020) Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. J Hydrol 583:124631. https://doi.org/10.1016/j.jhydrol.2020.124631

Kao I-F, Liou J-Y, Lee M-H, Chang F-J (2021) Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts. J Hydrol 598:126371. https://doi.org/10.1016/j.jhydrol.2021.126371

Kefi M, Mishra BK, Masago Y, Fukushi K (2020) Analysis of flood damage and influencing factors in urban catchments: case studies in Manila, Philippines, and Jakarta, Indonesia. Nat Hazards 104:2461–2487. https://doi.org/10.1007/s11069-020-04281-5

Kerh T, Lee C (2006) Neural networks forecasting of flood discharge at an unmeasured station using river upstream information. Adv Eng Softw 37:533–543. https://doi.org/10.1016/j.advengsoft.2005.11.002

Kratzert F, Klotz D, Brenner C, Schulz K, Herrnegger M (2018) Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrol Earth Syst Sci 22:6005–6022. https://doi.org/10.5194/hess-22-6005-2018

Kratzert F, Klotz D, Herrnegger M, Sampson AK, Hochreiter S, Nearing GS (2019a) Toward improved predictions in ungauged basins: exploiting the power of machine learning. Water Resour Res 55:11344–11354. https://doi.org/10.1029/2019WR026065

Kratzert F, Klotz D, Shalev G, Klambauer G, Hochreiter S, Nearing G (2019b) Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. Hydrol Earth Syst Sci 23:5089–5110. https://doi.org/10.5194/hess-23-5089-2019

Le X-H, Ho HV, Lee G, Jung S (2019) Application of long short-term memory (LSTM) neural network for flood forecasting. Water 11:1387. https://doi.org/10.3390/w11071387

Lei X et al (2021) Urban flood modeling using deep-learning approaches in Seoul, South Korea. J Hydrol 601:126684. https://doi.org/10.1016/j.jhydrol.2021.126684

Liu M et al (2020) The applicability of LSTM-KNN model for real-time flood forecasting in different climate zones in China. Water 12:440. https://doi.org/10.3390/w12020440

Nhita F, Saepudin D, Wisesty UN Comparative study of moving average on rainfall time series data for rainfall forecasting based on evolving neural network classifier. In: 2015 3rd International symposium on computational and business intelligence (ISCBI), 2015. IEEE, pp 112–116. doi:https://doi.org/10.1109/ISCBI.2015.27

Ni L, Wang D, Singh VP, Wu J, Wang Y, Tao Y, Zhang J (2020) Streamflow and rainfall forecasting by two long short-term memory-based models. J Hydrol 583:124296. https://doi.org/10.1016/j.jhydrol.2019.124296

Rahmati O, Darabi H, Haghighi AT, Stefanidis S, Kornejady A, Nalivan OA, Tien Bui D (2019) Urban flood hazard modeling using self-organizing map neural network. Water 11:2370. https://doi.org/10.3390/w11112370

Razavi S, Karamouz M (2007) Adaptive neural networks for flood routing in river systems. Water Int 32(360):375. https://doi.org/10.1080/02508060708692216

Sayers W, Savić D, Kapelan Z, Kellagher R (2014) Artificial intelligence techniques for flood risk management in urban environments. Proc Eng 70:1505–1512. https://doi.org/10.1016/j.proeng.2014.02.165

Shen H-Y, Chang L-C (2013) Online multistep-ahead inundation depth forecasts by recurrent NARX networks. Hydrol Earth Syst Sci 17:935. https://doi.org/10.5194/hess-17-935-2013

Siou LKA, Johannet A, Borrell V, Pistre S (2011) Complexity selection of a neural network model for karst flood forecasting: the case of the Lez Basin (southern France). J Hydrol 403:367–380. https://doi.org/10.1016/j.jhydrol.2011.04.015

Song T, Ding W, Wu J, Liu H, Zhou H, Chu J (2020) Flash flood forecasting based on long short-term memory networks. Water 12:109. https://doi.org/10.3390/w12010109

Wallemacq P, House R (2018) Economic losses, poverty and disasters: 1998–2017. Centre for Research on the Epidemiology of Disasters, Brussels, Belgium

Wang Y, Zhang J, Chang P-L, Langston C, Kaney B, Tang L (2016) Operational C-band dual-polarization radar QPE for the subtropical complex terrain of Taiwan. Adv Meteorol 2016:1–15. https://doi.org/10.1155/2016/4294271

Ward P, Marfai M, Yulianto F, Hizbaron D, Aerts J (2011) Coastal inundation and damage exposure estimation: a case study for Jakarta. Nat Hazards 56:899–916. https://doi.org/10.1007/s11069-010-9599-1

Xiang Z, Yan J, Demir I (2020) A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resour Res 56:e2019WR025326. https://doi.org/10.1029/2019WR025326

Yuan X, Chen C, Lei X, Yuan Y, Adnan RM (2018) Monthly runoff forecasting based on LSTM–ALO model. Stoch Environ Res Risk Assess 32:2199–2212. https://doi.org/10.1007/s00477-018-1560-y

Zhang D et al (2018a) Modeling and simulating of reservoir operation using the artificial neural network, support vector regression, deep learning algorithm. J Hydrol 565:720–736. https://doi.org/10.1016/j.jhydrol.2018.08.050

Zhang D, Lindholm G, Ratnaweera H (2018b) Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. J Hydrol 556:409–418. https://doi.org/10.1016/j.jhydrol.2017.11.018

Zhao G, Pang B, Xu Z, Peng D, Xu L (2019) Assessment of urban flood susceptibility using semi-supervised machine learning model. Sci Total Environ 659:940–949. https://doi.org/10.1016/j.scitotenv.2018.12.217

Zhu S, Luo X, Yuan X, Xu Z (2020) An improved long short-term memory network for streamflow forecasting in the upper Yangtze River. Stoch Env Res Risk Assess 34:1313–1329. https://doi.org/10.1007/s00477-020-01766-4

Zuo G, Luo J, Wang N, Lian Y, He X (2020) Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting. J Hydrol 585:124776. https://doi.org/10.1016/j.jhydrol.2020.124776

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Song-Yue Yang[1] · **Bing-Chen Jhong**[2] · **You-Da Jhong**[3] · **Tsung-Tang Tsai**[4] · **Chang-Shian Chen**[5]

[1] Department of Urban Planning and Spatial Information, Feng Chia University, Taichung 407102, Taiwan

[2] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan

[3] College of Engineering and Science, Feng Chia University, No. 100, Wenhua Rd. Xitun Dist., Taichung City 407102, Taiwan

[4] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407102, Taiwan

[5] Department of Water Resources Engineering and Conservation, Feng Chia University, Taichung 407102, Taiwan