# An optimized LSTM-based approach applied to early warning and forecasting of ponding in the urban drainage system

Wen Zhu[1], Tao Tao[1], Hexiang Yan[1], Jieru Yan[1], Jiaying Wang[1], Shuping Li[1], Kunlun Xin[1]

[1]College of Environmental Science and Engineering, Tongji University, Shanghai, 200000, China

*Correspondence to: Tao Tao (taotao@tongji.edu.cn), Jieru Yan (yan_jieru@tongji.edu.cn)*

**Abstract.** In this study we propose an optimized LSTM-based approach which is applied to early warning and forecasting of ponding in urban drainage system. This approach can quickly identify and locate ponding with relatively high accuracy. Based on the approach, a model is developed, which is constructed by two tandem processes and utilizes a multi-task learning mechanism. The superiority of the developed model was demonstrated by comparing with two widely used neural networks (LSTM, CNN). Then, the model was further revised with available monitoring data in the study area to achieve higher accuracy. We also discussed how the number of selected monitoring points influenced the performance of the corrected model. In this study, over 15000 designed rainfall events were used for model training, covering various extreme weather conditions.

## 1 Introduction

The intensity and frequency of urban floods are growing as a result of the increased frequency of extreme weather, rapid urbanization, and climate change (Hossain Anni et al., 2020; Guo et al., 2021; Huong and Pathirana, 2013). It is becoming increasingly clear that urban floods significantly impact city management and endanger the safety of people's life and property. The ability to reliably characterize and forecast urban floods and generate high-precision flood risk maps has become critical in flood mitigation and decision-making.

The most common approach to simulating urban floods is to develop a hydrodynamic model (i.e. storm-inundation simulation), which utilizes the collected topographic map, information on the pipe network, historical rainfall data, monitoring data, and other information in the study area (Jamali et al., 2018; Aryal et al., 2020; Balstrøm and Crawford, 2018; Tian et al., 2019). However, a realistic hydrodynamic model for continuous simulation requires vast data, such as comprehensive information on topography, infiltration conditions, and sewage system data (including exact locations, depths, and diameters of sewage pipes)., However, the above data are difficult to obtain, especially in metropolitan areas (Rahman et al., 2002; Kuczera et al., 2006). Furthermore, calculation in storm-inundation simulation is sophisticated, and often computationally intensive, which takes a long time to execute. The most detailed representation of the storm-inundation simulation is the 1D-2D model (Djordjević et al., 1999; Djordjević et al., 2005) which summarizes the dynamic interaction between the flow that enters the underground drainage network and the overloaded flow that spreads to the surface flow

network during high-intensity rainfall. Representatives of such model include XPSWMM, TUFLOW, and MIKE FLOOD (Leandro and Martins, 2016; Teng et al., 2017; Zhang and Pan, 2014).

The lack of underlying information has hampered the continuous development of hydrodynamic models in urban flood forecasting. As a result, deep learning has emerged as another viable forecasting tool. Deep learning is a particular machine learning technique that leverages neural networks to learn nonlinear relationships from a dataset (Mudashiru et al., 2021; Sit et al., 2020; Shen, 2018; Moy De Vitry et al., 2019). It can compensate for data scarcity by training on a large designed data set. Unlike traditional hydrodynamic models, deep learning does not require any assumptions on the physical processes behind it.

However, there are opportunities to further the application of deep learning in urban flood forecasting. Firstly, the training dataset needs to be enriched to reflect the superiority of the approach. Many studies in urban flood forecasting only use a small number of samples to develop the deep learning models. For example, Cai and Yu (2022) used 25 historical floods for forecasting. Abou Rjeily et al. (2017) used only 10 rainfall events for training and verification, which was insufficient to reflect the characteristics of rainfall distribution. Secondly, due to the high cost of monitoring equipment, researchers usually have to rely on unvalidated simulations produced from hydrodynamic models. For example, Chiang et al. (2010) used synthetic data from the SWMM model as the target values to train the recurrent neural network (RNN), then compared the predictions with simulation results to evaluate the model accuracy in estimating water levels at ungauged locations. Thirdly, some studies have focused on building more complex deep-learning architectures to improve model performance. Example includes but not limited to the automatic encoder (Bai et al., 2019), encoder-decoder (Kao et al., 2020c), and customized layers based on Long Short-Term Memory (LSTM) (Sit et al., 2020; Kratzert et al., 2019; Kratzert et al., 2019). For example, an encoder-decoder LSTM has been proposed for runoff forecasting up to 6 hours and 24 hours ahead (Xiang et al., 2020b; Kao et al., 2020c). Nevertheless, the urban flooding forecasting tasks with multiple time steps are mainly based on the precipitation forecast hours in advance, which is not available in this paper. With the real-time rain data in a short duration, getting enough data like the continuing runoff data to support the hours ahead prediction is yet to be available.

In this study, we propose an optimized LSTM-based approach for early warning and forecasting of ponding in urban drainage system. This approach can quickly identify and locate ponding with relatively high accuracy. The model is constructed by two tandem processes and introduces a multi-task learning mechanism. The evaluation results of the model were compared with those of two widely used neural networks, i.e., LSTM and CNN. The model was further revised with monitoring data in the study area to improve the emulation performance. We also discussed the influence of the number of monitoring points selected on the model performance. Over 15000 designed rainfall events were used for model training, covering various extreme weather conditions.

The rest of the paper is organized as follows: Section 2 introduces the methodology used to develop the LSTM-based modelling framework, as well as the experimental setup and application of the model. Sections 3 presents the results of the model. Section 4 presents discussion, Section 5 concludes this paper by drawing brief conclusions.

## 2 Methodology

### 2.1 LSTM-based model

Like a hydrodynamic model, which is generally composed of two processes: runoff process and flow confluence process, the LSTM-based model proposed in this study is also constructed with two stages. Fig. 1 illustrates the model architecture from input (i.e. rain intensity) to output (i.e. ponding volume of each node).

The two processes are in tandem: the inputs of the flow confluence process are inherited and concatenated from the outputs of all nodes in the runoff process. However, during the training process, the two processes are trained separately without mutual interference as the inputs and outputs of both processes are produced from a hydrodynamic model.

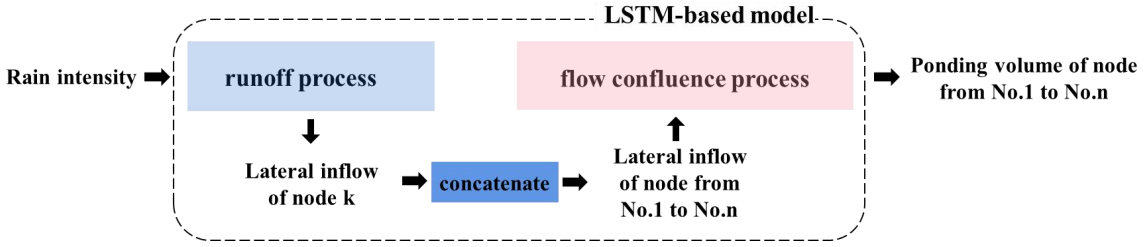



**Figure 1: The architecture of the LSTM-based model.**

#### 2.1.1 Runoff process

With a general understanding of a hydrodynamic model, the runoff process involves surface runoff and infiltration, while the most important influential factor is rainfall. As a mass rainfall curve can reflect the characteristics of a specific rainfall process, it can be directly used as the input of a neural network. The output of the neural network (i.e. lateral inflows at each node) reflects the hydraulic state in the runoff process.

Fig. 2 illustrates the training, validation, and testing procedures in the runoff process. As shown in Fig.2, a training set with two time-series data is fed into the neural network: rainfall intensity and lateral inflow at each node. At each epoch, four indicators are used to evaluate the consistency between the predicted lateral inflows and the simulation from the hydrodynamic model. If the model converges, the network is further evaluated on the test set. Otherwise, the next training epoch is started.

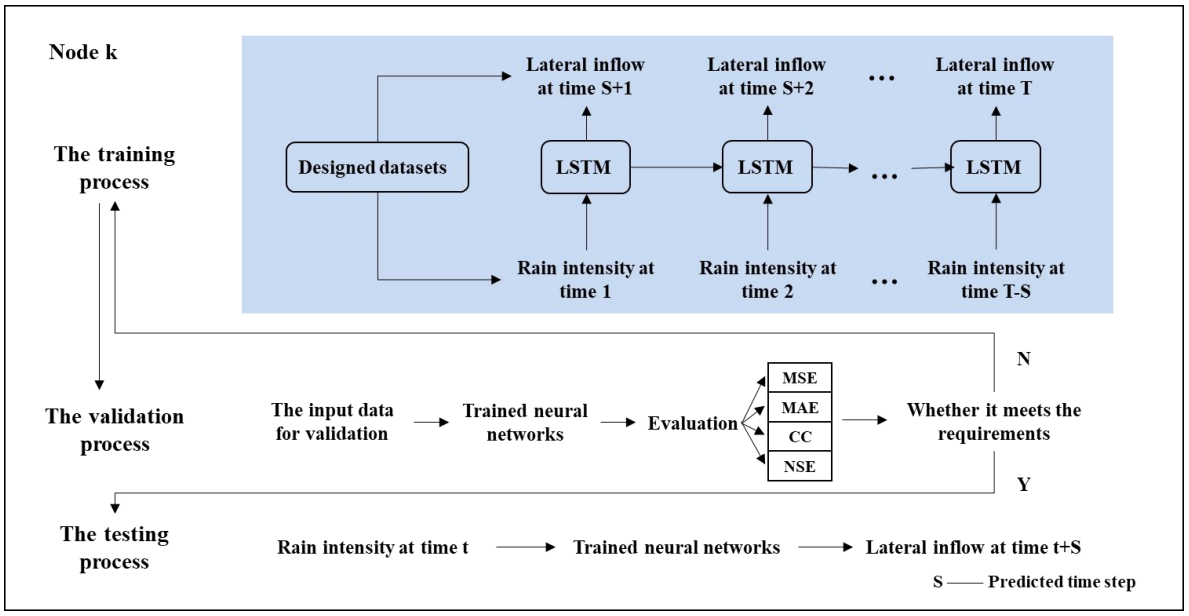



Figure 2: The training, validation, and testing procedures used when developing the LSTM-based runoff emulator. (MAE -- Mean Absolute Error, MSE -- Mean Squared Error, CC -- Correlation coefficient, NSE -- Nash-Sutcliffe efficiency coefficient)

### 2.1.2 Flow confluence process

The flow confluence process is set up in the same manner as the simulation process of a hydrodynamic model (e.g., the SWMM model). If we compare the urban drainage system to a black box, only the lateral inflows at each node and outflows from the outlets enter and leave the system, respectively (Archetti et al., 2011). If a free outflow condition is considered, namely the hydraulic state behind the outlets has little influence on the interior of the system, then the inputs of the flow confluence process are only the lateral inflows at each node. Fig. 3 illustrates the details of the network architecture in the flow confluence process.

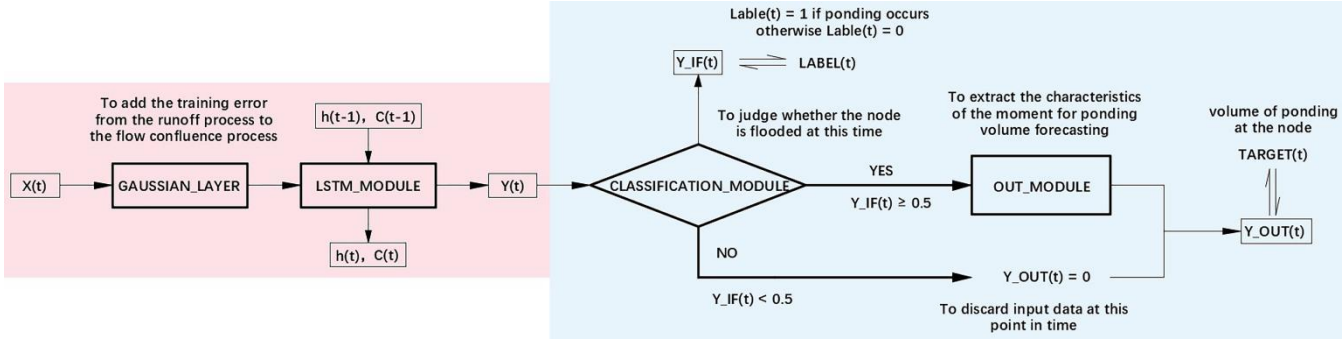



**Figure 3: The network structure of the flow confluence process (for a single node).**

As illustrated in the pink block in Figure 3, a Gaussian layer is added after the input layer in the flow confluence process during training. The gaussian layer serves as a filter to compensate for the inaccuracy of the prediction (by the hydrodynamic model) in the runoff process. The model is trained to minimize the differences between the predictions (from the neural network, i.e. the output from the runoff process) and the simulations (from the hydrodynamic model). Then (as illustrated in the blue block in Figure 3), a classification layer is added after the outputs of the LSTM module to judge whether ponding occurs at the time step. Only when ponding occurs at the time step, can the output of the LSTM module enters the 'OUT_MODULE' to continue with the learning. Otherwise, the output of the LSTM module at this time step is discarded. In this way, the interference of the time points without ponding on the ponding volume forecasting is eliminated to a great extent. The higher the classification accuracy is, the more accurate the prediction of ponding volume will be. Moreover, the multi-task learning has a hard parameter sharing mechanism, which effectively alleviate the over-fitting of the model. The parameters in the 'LSTM_MODULE' (including the parameters of the LSTM layers, batch normalization layers, activation functions, etc.) are shared by the 'CLASSIFICATION_MODULE' and 'OUT_MODULE'.

### 2.2 Error transmission

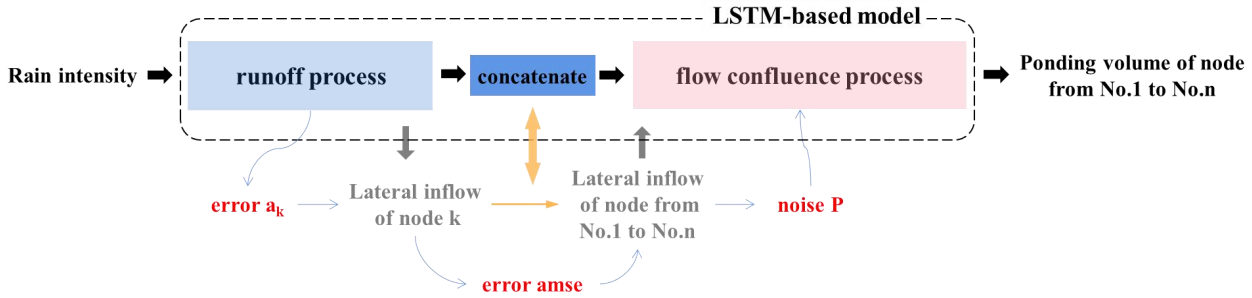



**Figure 4: The error transmission during training from the runoff process to the flow confluence process.**

Fig. 4 illustrates how the training error of the neural network propagates from the runoff process to the flow confluence process during training. Noise P ($P \sim N(0, p^2)$) is added to the lateral inflows before feeding the data into the neural network in the flow confluence process, in order to avoid the interference caused by the training error in the runoff process and also to alleviate the over-fitting of the neural network. The magnitude of noise P can be determined as follows:

1 The mean squared error (MSE) is used to characterize the training error in the runoff process, where the error at node K can be computed by

$$a_k = \frac{\sum_{i=1}^{T}\sum_{j=1}^{S}\left(\hat{X}_{ij} - X_{ij}\right)^2}{T \cdot S} \tag{1}$$

where T represents the duration of event j in min, S represents the number of events in the training data, $\hat{X}_{ij}$ represents the simulated lateral inflow at node K at the i-th time step in the j-th rainfall event in $L \cdot s^{-1}$, $X_{ij}$ represents the output of the runoff process at node K at the i-th time step in the j-th sample event in $L \cdot s^{-1}$.

2 Then compute the average mean squared error of all nodes by

$$\mathrm{amse} = \frac{\sum_{k=1}^{N} a_k^2}{N} \tag{2}$$

where N represents the number of nodes.

3 Then convert amse into noise percentage ε with the mean value of the predicted lateral inflows at all nodes in the training set by

$$\varepsilon = \sqrt{\frac{P_N}{P_S}} = \sqrt{\frac{\sum_{k=1}^{N}\sum_{i=1}^{T}\sum_{j=1}^{S}\left(\hat{X}_{kij}-X_{kij}\right)^2}{\sum_{k=1}^{N}\sum_{i=1}^{T}\sum_{j=1}^{S}\left(X_{kij}\right)^2}} \le \sqrt{\frac{\sum_{k=1}^{N}\sum_{i=1}^{T}\sum_{j=1}^{S}\left(\hat{X}_{kij}-X_{kij}\right)^2}{\frac{1}{N\cdot T\cdot S}\left(\sum_{k=1}^{N}\sum_{i=1}^{T}\sum_{j=1}^{S}X_{kij}\right)^2}} = \frac{\sqrt{\mathrm{amse}}}{\frac{1}{N\cdot T\cdot S}\sum_{k=1}^{N}\sum_{i=1}^{T}\sum_{j=1}^{S}X_{kij}} \tag{3}$$

where $P_S$ represents signal power, and $P_N$ represents noise power.

4 Finally add noise P to the inputs (X) in the flow confluence process during the training process. Namely, generate a set of random numbers G with the length of X using Pseudorandom Number Generator, where G obeys a normal distribution (G~N(0,1)), i.e., $P = p \cdot G$, where p is computed by

$$p = \varepsilon \cdot \sqrt{\frac{1}{T}\sum_{i=1}^{T}(X_i)^2} \tag{4}$$

### 2.3 Model correction system

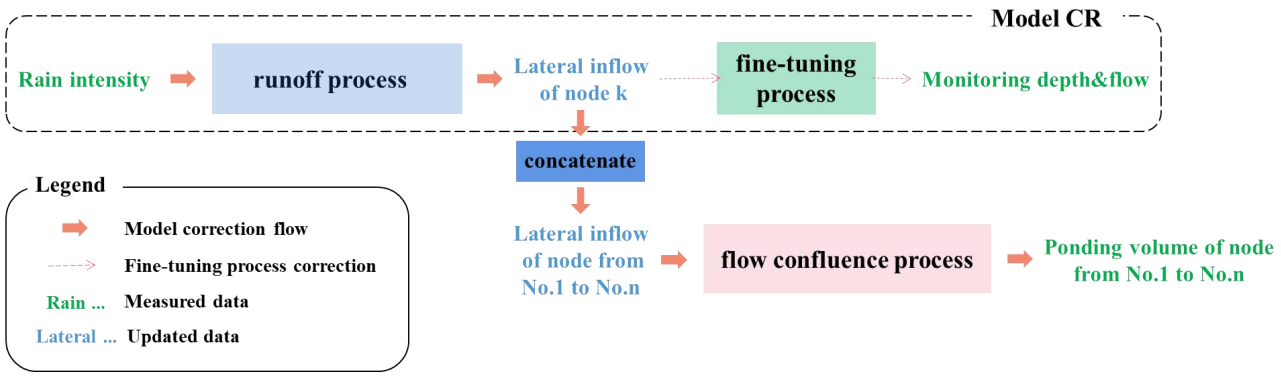



**Figure 5: Model correction system. CR is abbreviated for correction of the runoff process.**

The LSTM-based model is built based on the simulation results of a relatively accurate hydrodynamic model. However, the differences between the simulation from the hydrodynamic model at the monitoring points and the obtained monitoring data always exist during the operation of the pipe network, which leads to a discrepancy between the predicted results from the proposed LSTM-based model and the actual situation. Thus it is necessary to correct the model using the measured level and flow data at the monitoring points. Moreover, how to revise the model properly using the available data is also one of the focuses of this study. Fig. 5 describes the model correction process using the measured rainfall data, depths and flows at the monitoring points, and ponding data at any node. Specifically, the LSTM-based model is corrected using the following two steps:

1. The runoff process is corrected with the measured rain, level, and flow data referring to the transfer learning. Transfer learning is mainly used to transfer the knowledge of one domain (source domain) to another domain (target domain) such that the target domain can achieve better learning effects (PAN, S J, et al., 2010).

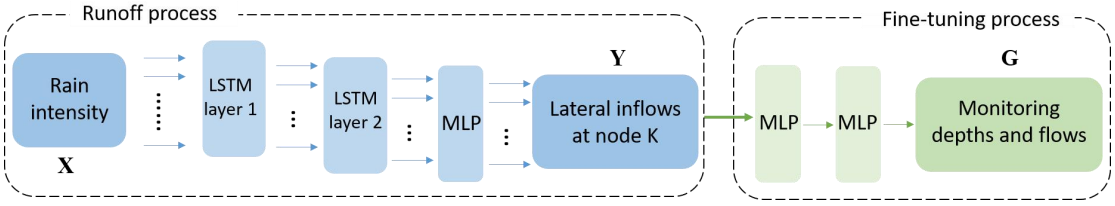



**Figure 6: The architecture of Model CR. (MLP -- Multi-Layer Perception)**

Fig. 6 shows the schematic of Model CR. It migrates the network structure of the runoff process from rain data (X) to lateral inflows (Y) to the input-output connection between X and monitoring data (G). Then, multiple fully connected layers are added after the output layer of Y. Model CR is designed to update the runoff process in the primary LSTM-based model. The correction has two steps: training and updating. Firstly, the model CR is trained based on a pre-trained mapping from X to Y (as shown in Section 2.1.1) with constructed rain data, simulated level and flow data. Then, it is updated on pairs of measured rain data, monitored water depths and flows.

2. The flow confluence process is corrected using the updated lateral inflows of all concatenated nodes and the measured ponding volume.

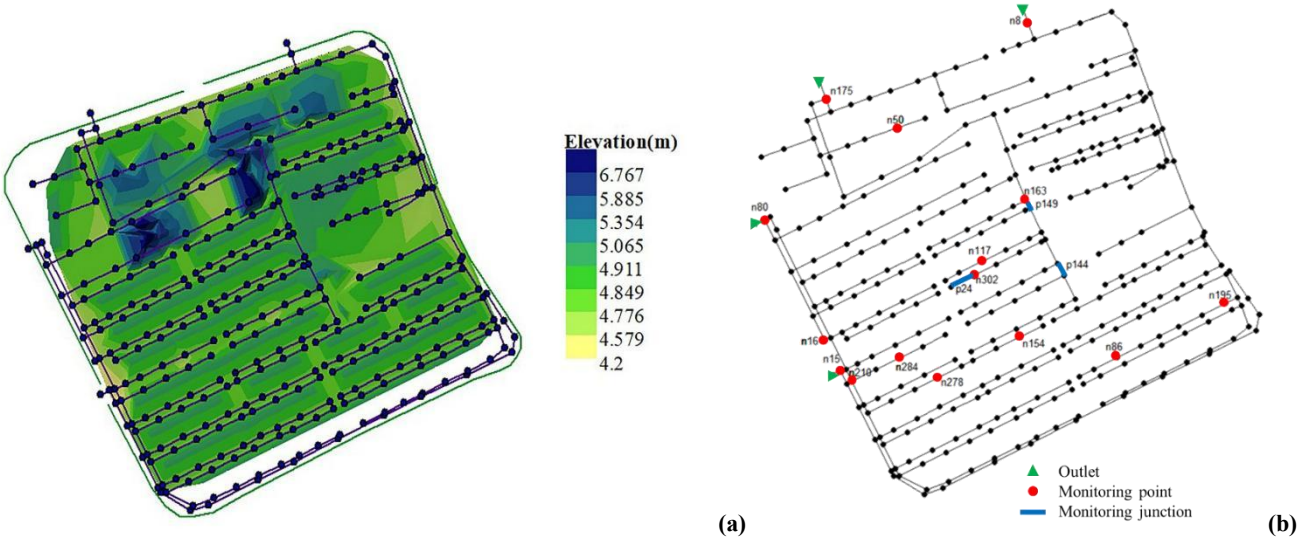
### 2.4 Case study

#### 2.4.1 Study area

The LSTM-based model trains nodes in the pipe network one by one. Namely N sub-models with the same architecture are generated, where N is the number of nodes in the system. Both in the runoff process and flow confluence process, these sub-models are trained separately. Due to this structural characteristic, the size of the case area does not limit the model's performance. Regarding the model structure, the output of the runoff process is the lateral inflow at a single node. Likewise, the output of the flow confluence process is the ponding volume at a single node. Regardless of the size of the pipe network, the output of the model is at each node. However, a large-scale pipe network with lots of nodes will significantly increase the time spent training the model and also require extra processing power.

To verify the feasibility of the modelling framework above, a small-scale case area JD, a residential district in S city, is selected as the study area. Fig. 7(a) shows the elevation map of the study area. There are 32 residential buildings in the district, with a total area of 6.128hm$^2$. The study area is separated from the municipal roads by walls with three entrances on the community's north, east, and west sides. Rain pipes in the study area are circular pipes with 200mm, 300mm, 400mm, 500mm, or 600mm in diameter (mostly 300mm). The total length of this pipe network is 5.5 km. The network contains 336 nodes and 340 pipes, and is connected to the municipal pipe networks through 4 outlets, as denoted by the green triangle in

Fig. 7(b). There are 15 level gauges and three flowmeters in the current pipe network. The layout of monitoring points is also shown in Fig. 7(b).




(a) (b)

**Figure 7: Study area - JD residential district. (a) The elevation map and stormwater system in the case area. (b) The layout of monitoring points in the case area.**

### 2.4.2 Rainfall data

The rainstorm intensity for S city is designed using Eq. (5), which is obtained according to a universal design storm pattern proposed by Keifer&Chu. The storm pattern is broadly used both at home and abroad. The generated storms are usually extreme enough to reflect the state of the pipe networks under the most unfavorable conditions (Skougaard Kaspersen et al., 2017).

$$q = \frac{167A(1 + C\log P)}{(t+b)^n} = \frac{1600(1 + 0.846\log P)}{(t+7.0)^{0.656}} \tag{5}$$

where q is the rainstorm intensity in $L \cdot s^{-1} \cdot hm^{-2}$ P is the reappearing rainfall period in a, t is the duration of rainfall in min, A, C, b, and n are parameters of the rainstorm intensity design formula.

The rainstorm intensity before or after the peak is determined using Eq. (6).

$$\begin{cases} I(t_b) = \dfrac{A(1 + C\lg P)\left[\dfrac{(1-n)}{r}t_b + b\right]}{\left(\dfrac{t_b}{r} + b\right)^{n+1}} \\ I(t_a) = \dfrac{A(1 + C\lg P)\left[\dfrac{(1-n)}{1-r}t_a + b\right]}{\left(\dfrac{t_a}{1-r} + b\right)^{n+1}} \end{cases} \tag{6}$$

where $t_b$ and $t_a$ are the time before and after the peak in min, respectively, r is the rainfall peak coefficient.

Then single-peak rainfall scenarios were constructed unevenly by using different rainfall reappearing periods (P) ranging from 0.5a to 100a, peak coefficient (r) ranging from 0.1 to 0.9, and duration (T) ranging from 60 to 360 min.

In addition to single-peak rainfall scenarios, we also considered bimodal rainfall scenarios. According to the historical bimodal rainfall data in S city, the rainfall peaks corresponding to the bimodal design storm pattern with the duration from 60 to 360 min could be computed by Pilgrim & Cordery. Pilgrim & Cordery is a method to count the historical rainfall data and deduce the rainstorm pattern from it (Pilgrim & Cordery, 1975).

**Table 1: The bimodal design storm pattern.**

**(a) 60 minutes duration**

| T(5min) | P/Pmax(%) |
|---|---|
| 1 | 3.19 |
| 2 | 16.71 |
| 3 | 8.74 |
| 4 | 1.52 |
| 5 | 2.27 |
| 6 | 4.02 |
| 7 | 5.89 |
| 8 | 22.68 |
| 9 | 10.51 |
| 10 | 12.76 |
| 11 | 7.05 |
| 12 | 4.68 |

**(b) 120 min duration**

| T(5min) | P/Pmax(%) |
|---|---|
| 1 | 0.74 |
| 2 | 1.76 |
| 3 | 6.42 |
| 4 | 3.75 |
| 5 | 2.05 |
| 6 | 1.52 |
| 7 | 2.73 |
| 8 | 4.43 |
| 9 | 9.23 |

| T(5min) | P/Pmax(%) |
|---|---|
| 10 | 11.17 |
| 11 | 17.57 |
| 12 | 13.81 |
| 13 | 7.76 |
| 14 | 5.3 |
| 15 | 2.38 |
| 16 | 1.13 |
| 17 | 3.17 |
| 18 | 1.32 |
| 19 | 0.98 |
| 20 | 0.84 |
| 21 | 0.54 |
| 22 | 0.64 |
| 23 | 0.44 |
| 24 | 0.33 |

Table 1 shows the bimodal design storm patterns with 60 min and 120 min duration time, respectively, where P/Pmax represents the distribution of rainfall intensity over time (with 5 min unit period). Then, double-peak rainfall scenarios were constructed according to Table 1 using reappearing periods ranging from 0.5a to 100a.

The produced single-/double-peak rainfall data were then added with Gaussian white noise (produced according to the procedures described in Section 2.2) to ensure that the obtained dataset contains enough extreme conditions. Take the rainfall with a return period of 5a as an example. Fig. 8 shows the effect of adding noise, where the subfigure (1) shows the randomly generated Gaussian white noise over the duration, the subfigure (2) shows the distribution of reordered white noise, and the subfigure (3) zooms in on the part circled in (2). The subfigures (4) - (6) show the design rainfalls after adding 30%, 50%, and 70% white noise, respectively. Specifically, we have limited the noises near the rainfall peak, i.e., only negative noises are allowed there.

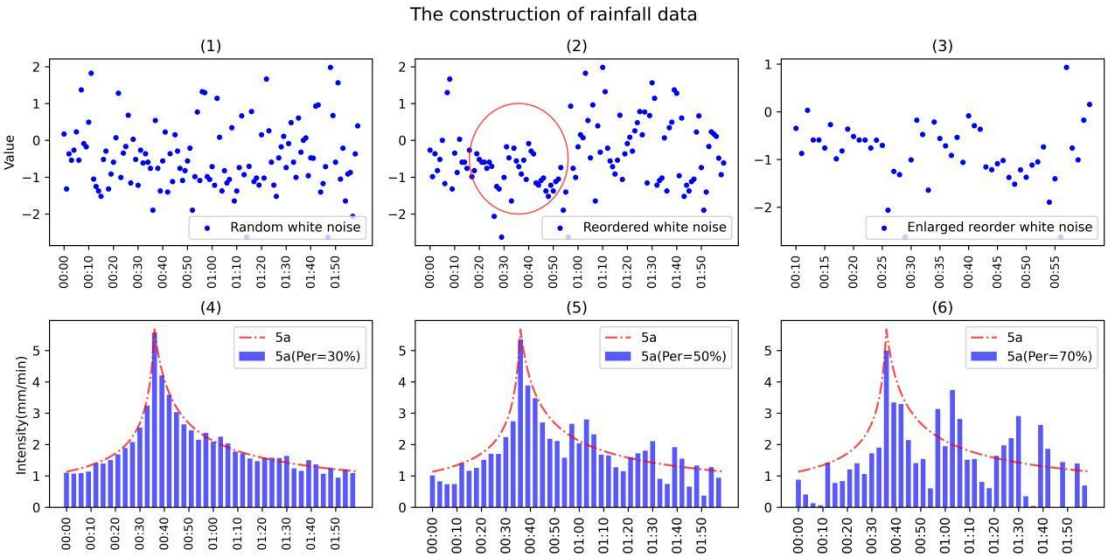



**Figure 8: A demonstrative example to show the effect of adding white noise.**

In this study, the noise percentages went from 0 to 100% with an increment of 10% to blur the characteristics of the design storm pattern and intensify the extreme conditions. The synthetic dataset contained a total of 16960 rainfall events. The ratios of the training, validation, and test sets were 80%, 10%, and 10%, respectively.

In general, a small training set normally leads to poor approximation effect. Thus a convergence test was performed to evaluate how much data was required for the proposed LSTM-based model to obtain the desired approximation effect. The model performances using different sizes of training data were compared, as shown in Fig. 9. When the data size was reduced to 2/3 of the origin volume, the model performance fell down to 90% of the original. Moreover, if the data size was halved, less than 80% of the origin model performance remained.

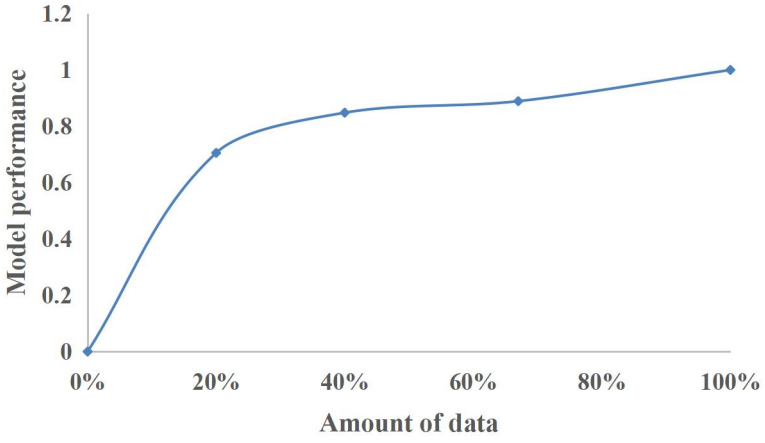



**Figure 9: The learning curve which describes the relationship between model performance and data volume.**

### 2.4.3 Simulated and measured data

A hydrodynamic model was established for the case pipe network. The simulation results (i.e., the lateral inflows and the volume of ponding at each node, as well as the level and flow data at the monitoring points) were obtained using the constructed rainfall events described in Sect. 2.4.2. In the simulation process, we considered a uniform rainfall distribution in space. A simplified representation of the sewer system and a constant, uniform infiltration rate in the green area were also considered for runoff computation (Roland L¨owe et al.,2021). Meanwhile, we did not consider the two-dimensional surface overflow.

Besides, the measured rain data and monitoring data (water depth and flow) of five historical rainfall events were used to verify the performance of the corrected model. The uncertainty of the measurements was not considered (Huong and Pathirana, 2013). In this study, we considered the simulation results of the verified hydraulic model as the ground truth.

Table 2 shows the measurements of the five historical rainfall events used in the process of model correction. Among the five events, three were used to correct Model CR and the flow confluence process, the other two were used to evaluate the reliability of the approach.

**Table 2: 5 historical rainfall data used to correct the LSTM-based model.**

| Dataset | Rainfall event | Rainfall (mm) | Max. rain intensity (mm/min) | Duration (min) |
|---|---|---|---|---|
| Training | No.1 | 494.50 | 8.13 | 180 |
| | No.2 | 146.63 | 2.61 | 90 |
| | No.3 | 254.51 | 3.61 | 240 |
| Testing | No.4 | 442.61 | 7.26 | 150 |
| | No.5 | 254.41 | 4.97 | 120 |

### 2.4.4 Model Construction

**Table 3: Hyper-parameters configuration in the model setup and correction processes.**

| | Hyper-parameters | Model CR | | Flow confluence process |
|---|---|---|---|---|
| | | Runoff process | Fine-tuning process | |
| Model setup | Normalization | Z-score | Z-score | Min-Max |
| | Batch size | 150 | 150 | 150 |
| | Epoch | 300 | 300 | 300 |
| | Learning rate | 1e-2 | 5e-3 | 1e-2 |
| | Optimizer | Adam | SGD | SGD |

| Hyper-parameters | | Model CR | | Flow confluence process |
|---|---|---|---|---|
| | | Runoff process | Fine-tuning process | |
| Model correction | LSTM hidden layer neurons | 16 | - | 256 |
| | MLP hidden layer neurons | 16 | 1536/3072 | 256/128* |
| | LSTM layers | 2 | - | 4 |
| | MLP layers | 1 | 2 | 2* |
| | Learning rate | 1e-4 | | 5e-5 |
| | Optimizer | Adam | | SGD |

Note: * means to set the hyperparameters of 'CLASSIFICATION_MODULE' and 'OUT_MODULE' in the flow confluence process to the same values.

The hyper-parameters used in this paper were mainly determined by Hyperopt (BERGSTRA, J et al., 2013). Hyperopt is a Python library for hyper-parameter optimization that adjusts parameters using Bayesian optimization.

Table 3 shows the hyper-parameters in the learning process of the model setup and model correction obtained by Hyperopt.

### 2.4.5 Performance evaluation

Mean Absolute Error (MAE), Mean Squared Error (MSE), Correlation coefficient (CC) and Nash-Sutcliffe efficiency coefficient (NSE) are broadly used indicators to assess the performance of a data-driven model. In this study, we used MAE and MSE to quantify the size of the errors, i.e., difference at each node between the prediction by the proposed LSTM-based model and simulation from the hydrodynamic model. Moreover, NSE and CC were also used to evaluate the level of agreement at all nodes. Equations (7)-(10) list the formulas of these 4 indicators.

$$MAE = \frac{1}{DT}\sum_{s=1}^{D}\sum_{t=1}^{T}\left|Y_{st} - \hat{Y}_{st}\right| \tag{7}$$

$$MSE = \frac{1}{DT}\sum_{s=1}^{D}\sum_{t=1}^{T}\left(Y_{st} - \hat{Y}_{st}\right)^2 \tag{8}$$

$$NSE = 1 - \frac{\sum_{t=1}^{T}\left(\frac{1}{D}\sum_{s=1}^{D}Y_{st} - \frac{1}{D}\sum_{s=1}^{D}\hat{Y}_{st}\right)^2}{\sum_{t=1}^{T}\left(\frac{1}{D}\sum_{s=1}^{D}\hat{Y}_{st} - \frac{1}{DT}\sum_{t=1}^{T}\sum_{s=1}^{D}\hat{Y}_{st}\right)^2} \tag{9}$$

$$CC = \frac{\sqrt{\sum_{t=1}^{T}\left(\frac{1}{D}\sum_{s=1}^{D}Y_{st} - \frac{1}{DT}\sum_{t=1}^{T}\sum_{s=1}^{D}Y_{st}\right)\left(\frac{1}{D}\sum_{s=1}^{D}\hat{Y}_{st} - \frac{1}{DT}\sum_{t=1}^{T}\sum_{s=1}^{D}\hat{Y}_{st}\right)}}{\sqrt{\sum_{t=1}^{T}\left(\frac{1}{D}\sum_{s=1}^{D}Y_{st} - \frac{1}{DT}\sum_{t=1}^{T}\sum_{s=1}^{D}Y_{st}\right)^2}\sqrt{\sum_{t=1}^{T}\left(\frac{1}{D}\sum_{s=1}^{D}\hat{Y}_{st} - \frac{1}{DT}\sum_{t=1}^{T}\sum_{s=1}^{D}\hat{Y}_{st}\right)^2}} \tag{10}$$

where D is the number of events in the test set, T is the number time steps of the relevant rainfall event, $Y_{st}$ is the prediction given by the neural network at the t-th time step in the s-th event, $\hat{Y}_{st}$ is the simulation given by the hydrodynamic model.

To evaluate the accuracy of the proposed model in predicting ponding, we have introduced 5 indicators (as shown in Table 4): Accuracy (ACC), Precision (PPV), and False Omission Rate (FOR) to evaluate the model accuracy in predicting the occurrence of ponding at a single node; $S-PPV$ and $S-FOR$ to evaluate the model accuracy in predicting the occurrence of ponding for a single event.

**Table 4: Indicators used to evaluate accuracy of the proposed model in predicting ponding for a single node and for a single event.**

| Score | Purpose | Equation | Range | Best Value |
|---|---|---|---|---|
| ACC | Mean accuracy for time points classified correctly | $\mathrm{ACC}=\frac{1}{\mathrm{D}}\sum_{s=1}^{\mathrm{D}}\frac{\mathrm{TP_s}+\mathrm{TN_s}}{\mathrm{TP_s}+\mathrm{TN_s}+\mathrm{FP_s}+\mathrm{FN_s}}$ | $0-1$ | 1 |
| PPV | Mean precision for well-judged time point | $\mathrm{PPV}=\frac{1}{\mathrm{D}}\sum_{s=1}^{\mathrm{D}}\frac{\mathrm{TP_s}}{\mathrm{TP_s}+\mathrm{FP_s}}$ | $0-1$ | 1 |
| FOR | Mean proportion of omission on the timeline | $\mathrm{FOR}=\frac{1}{\mathrm{D}}\sum_{s=1}^{\mathrm{D}}\frac{\mathrm{FN_s}}{\mathrm{TN_s}+\mathrm{FN_s}}$ | $0-1$ | 0 |
| $S-PPV$ | Percentage for well-judged samples | $S-\mathrm{PPV}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$ | $0-1$ | 1 |
| $S-FOR$ | Percentage of samples with false negatives | $S-\mathrm{FOR}=\frac{\mathrm{FN}}{\mathrm{TN}+\mathrm{FN}}$ | $0-1$ | 0 |

where TP and TN denote the number of occurrences when a ponding case and a normal case (no ponding occurs) are correctly identified, respectively, FP is the number of occurrences when a normal case is incorrectly identified as a ponding case, FN is the number of occurrences when a ponding case is ignored by the model. The subscript "s" denotes the number of time steps in the s-th event.

## 3 Results

### 3.1 Model setup

The LSTM-based model was trained by the designed rainfall data and simulation produced from the hydrodynamic model. According to the procedures described in Sect. 2.2, the noise (ε) transmitted from the runoff process to the flow confluence process was equal to 1.9412% in the case pipe network. For the sake of convenience, the noise was set to 2%.

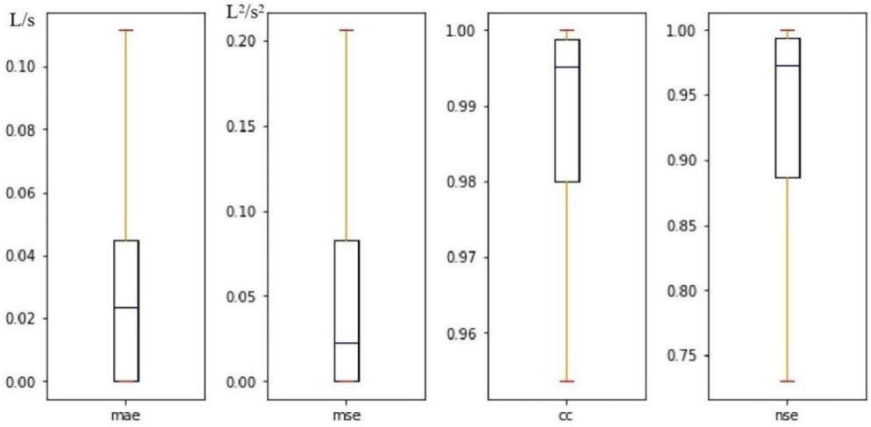



**Figure 10: Box plots of score values for comprehensive evaluation of all nodes in the case area in the model development procedure.**

Figure 10 described the overall performance of the model using four box plots of mean scores (of all nodes) on the test set, with the outliers removed. As shown in the figure, the median values of MAE and MSE were much smaller than 0.1, indicating that the model has converged at all nodes. The median value of CC was close to 1, even the minimum value was higher than 0.95. The median value of NSE was higher than 0.95, yet the minimum value was about 0.75, which indicated that although the model's performance at each node was slightly different, the overall prediction was generally reliable.

Due to the limited space, we only listed the evaluation results of 6 representative nodes. The six nodes (as shown in Fig. 11) were selected because of the severity of consequence once ponding occurred, and also because they were relatively uniformly distributed in the pipe network. Moreover, three of them (Nodes 2, 238, and 313) were chosen because the positive samples (where ponding occurred) accounted for less than 50% of the training set, and the other three were in the opposite case. For example, at Node 238, the positive samples accounted for 18.33% of the training set, while at Node 95, up to 98.6% samples were positive.

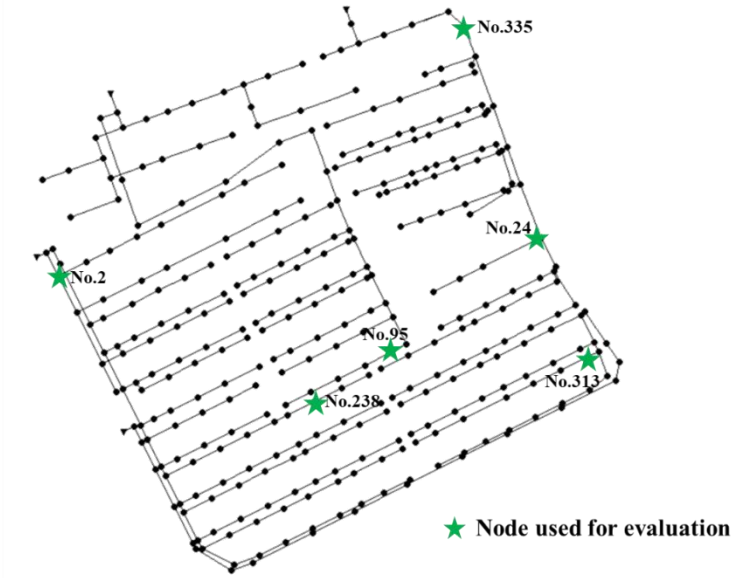



Figure 11: The locations of the six selected nodes used for evaluation.

**Table 5: The scores at the six selected nodes for evaluating the model performance in ponding occurrence prediction.**

| Node No. | ACC | PPV | FOR | S − PPV | S − FOR |
|---|---|---|---|---|---|
| 2 | 99.90% | 95.11% | 0.04% | 98.00% | 0.00% |
| 24 | 99.56% | 95.21% | 0.27% | 99.33% | 0.00% |
| 95 | 98.66% | 93.47% | 0.98% | 100.00% | 0.00% |
| 238 | 99.81% | 88.33% | 0.06% | 94.00% | 0.00% |
| 313 | 99.67% | 95.75% | 0.19% | 99.33% | 0.00% |
| 335 | 99.56% | 95.29% | 0.23% | 100.00% | 0.00% |

Table 5 lists the scores at the six selected nodes for evaluating the model performance in ponding occurrence prediction. In the table, Column ACC to FOR reflect the accuracy of ponding occurrence prediction in the sense of time (i.e. averaged in time). The mean ACC values (Accuracy) for all the 6 nodes were higher than 98.5%. Compared to ACC, the mean PPV values (Precision) were slightly lower, with the minimum value about 88% at Node 238, which indicated that a ponding case had at least 88% chance to be correctly identified. The mean FOR value (False Omission Rate) of each node was generally lower than 1%, among them the worst performance occurred at Node 95 (FOR=0.98%), which indicated that the model had a relatively small chance to ignored ponding. The last two columns in Table 5 reflect the accuracy of the model in predicting ponding occurrence for a single rainfall event. For example, falsely reported events took up 6% of the testing events at Node 238 (S-PPV=94%), which was already the worst performance among the 6 selected nodes. While the S-FOR values at all these 6 nodes were zero, which indicated that the model did not miss any ponding incidents in the testing set.

The scores for evaluating the model performance in ponding volume prediction are listed in Table 6. As shown in the table, the MAE and MSE scores were generally small, with the highest MAE score ($0.0770 \mathrm{L} \cdot \mathrm{s}^{-1}$) occurred at Node 95 and the highest MSE score ($0.3788 \mathrm{L}^2 \cdot \mathrm{s}^{-2}$) occurred at Node 2. Compared to the MAE and MSE scores, the variability of CC scores was much smaller. All of them were very close to 1. As for the NSE scores, the lowest score (NSE=0.8195 at Node 238) was above 0.8. The results shown in Table 6 indicated that the proposed model had a relatively good performance in ponding volume prediction.

**Table 6: The scores at the six selected nodes for evaluating the model performance in ponding volume prediction.**

| Node No. | MAE($\mathrm{L} \cdot \mathrm{s}^{-1}$) | MSE($\mathrm{L}^2 \cdot \mathrm{s}^{-2}$) | CC | NSE |
|---|---|---|---|---|
| 2 | 0.0170 | 0.3788 | 0.9997 | 0.9811 |
| 24 | 0.0414 | 0.1876 | 0.9941 | 0.9754 |
| 95 | 0.0770 | 0.2740 | 0.9860 | 0.9599 |
| 238 | 0.0073 | 0.0260 | 0.9999 | 0.8195 |
| 313 | 0.0183 | 0.0505 | 0.9974 | 0.9825 |
| 335 | 0.0349 | 0.0826 | 0.9968 | 0.9882 |

Furthermore, in the above analysis, mean score values on the test set were used for evaluation, and the variability was ignored. Figure 12 shows the predicted ponding volume at the selected nodes compared with the simulation results in six testing rainfall events. As shown in Subfigure (1), the predicted start time of ponding was 5 minutes earlier than the simulation at Node 2. As shown in Subfigure (2), three peaks appeared in the ponding process of Node 95, and the model has identified each of them. No ponding occurred at node 238 given the testing precipitation, as shown in Subfigure (5), and the prediction of the model was in consistent with it. Overall, the prediction of the model was relatively accurate.

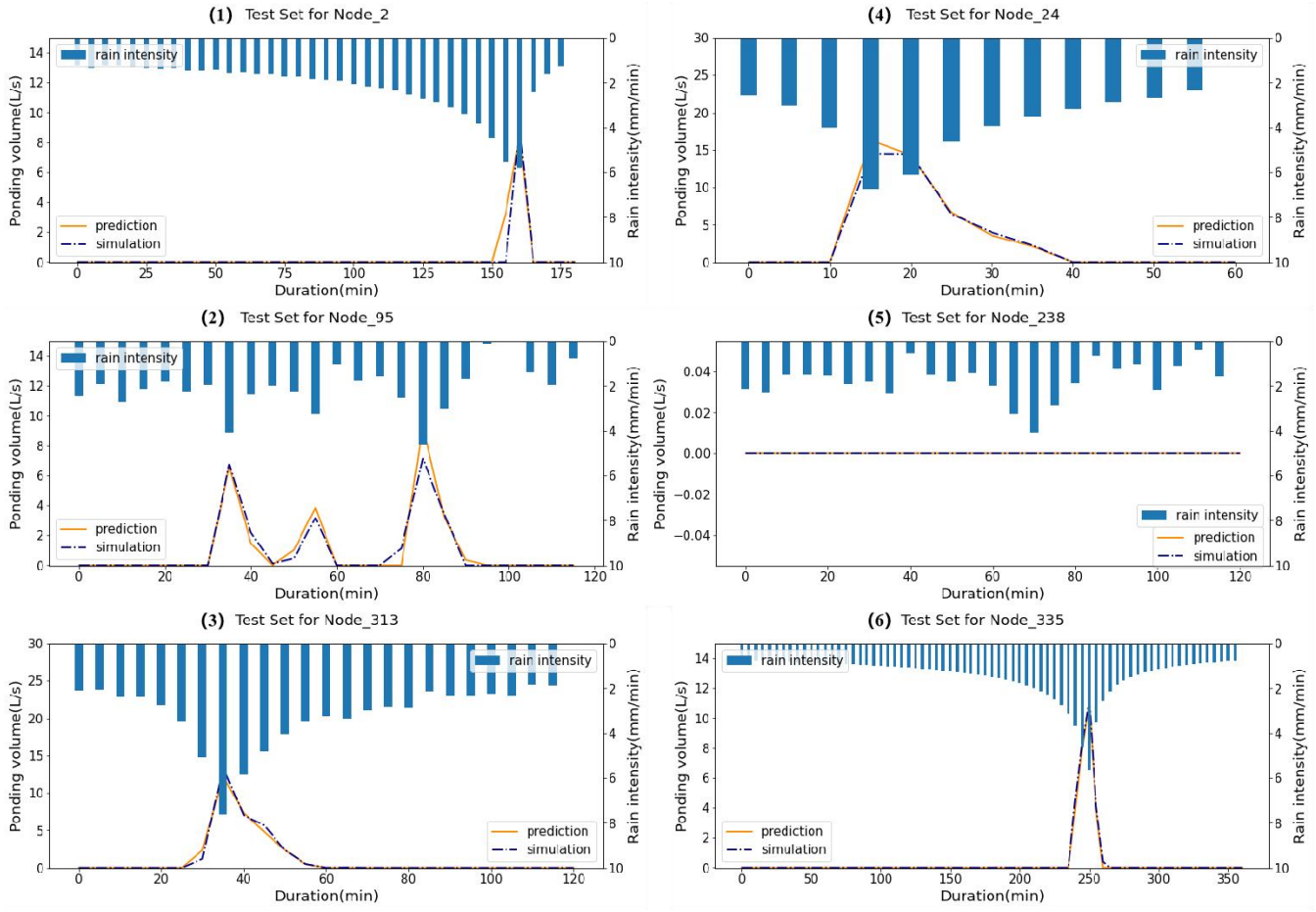



**Figure 12: Comparison between the predicted ponding volume and simulation from the hydrodynamic model at the selected nodes in 6 testing rainfall events that were chosen randomly.**

### 3.2 Model correction

In this study, the model was trained based on the simulation results from a hydrodynamic model. Though the hydrodynamic model has been verified, the differences between the simulation (from the hydrodynamic model) at the monitoring points and the monitoring data persisted during the essential operation of the pipe network, which inevitably degraded the accuracy of the LSTM-based model in ponding forecast. Thus, it is necessary to correct the model using the measured rainfall data, level or flow data at the monitoring points, and ponding data.

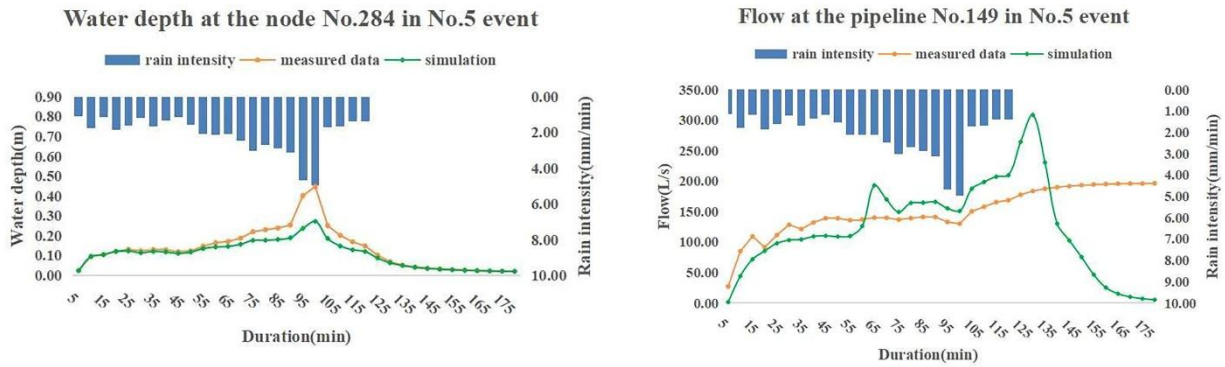



**Figure 13: Comparison between the measured data and simulation from the hydrodynamic model at Node 284 and Pipeline 149 in No.5 event.**

The discrepancy between the measurements and simulation from the hydrodynamic model can be exemplified by Fig. 13. As shown in the figure, rainfall event No.5 was one of the measured precipitation event where the maximum precipitation intensity reached 4.97mm/min. For this event, the measured water depth and flow data were compared with the simulation from the hydrodynamic model, as shown on the left and right panels, respectively, in Fig. 13.

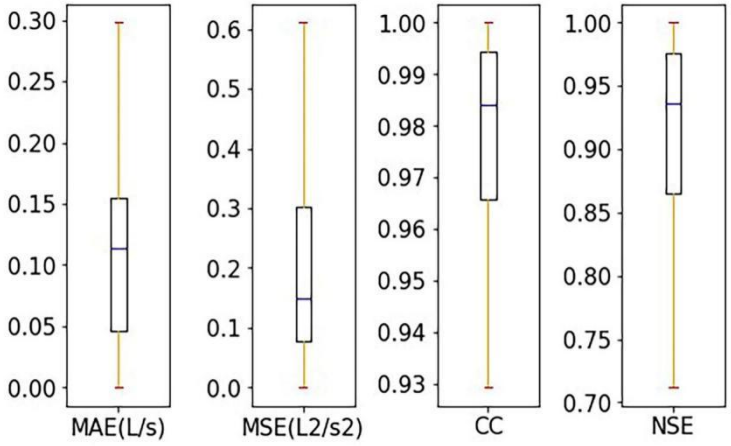



**Figure 14: Box plots of mean score values on the test set for all nodes in the model updating procedure.**

The ponding process predicted by the corrected model was compared with the monitored ponding data to evaluate the model performance. Figure 14 illustrates the overall performance of the corrected model using 4 indicators as described in Section 2.4.5. In specific, the 4 box plots show the range of the mean score values on the test set at all nodes. As shown in the figure, the median values of CC and NSE scores maintained above 0.98 and 0.9, respectively. In contrast, the maximum values of MAE and MSE scores remained lower than $0.30\mathrm{L}\cdot\mathrm{s}^{-1}$ and $0.6\mathrm{L}^2\cdot\mathrm{s}^{-2}$, respectively.

Specifically, the mean score values at the six selected nodes obtained using the corrected model are summarized in Table 7. As shown in the table, the MAE and MSE scores were generally small, the NSE score at each node was stably above 0.9, the

CC scores were all above 0.95. The results shown in Table 7 suggested that the corrected model performed well at different nodes.

**Table 7: Mean score values at the six selected nodes obtained using the corrected model.**

| Node No. | MAE($L \cdot s^{-1}$) | MSE($L^2 \cdot s^{-2}$) | CC | NSE |
|---|---|---|---|---|
| 2 | 0.0912 | 0.1604 | 0.9774 | 0.9545 |
| 24 | 0.1359 | 0.2426 | 0.9863 | 0.9586 |
| 95 | 0.2916 | 0.7378 | 0.9583 | 0.9071 |
| 238 | 0.0855 | 0.1842 | 0.9773 | 0.9302 |
| 313 | 0.0642 | 0.0571 | 0.9896 | 0.9727 |
| 335 | 0.0943 | 0.1135 | 0.9942 | 0.9683 |

To test further the capability of the corrected model, the mean scores of all nodes for five measured rainfall events are summarized in Table 8, where the results from the model without correction are also listed as a comparison. As shown in Table 8, all of the four indicators suggested that the corrected model performed much better than the model without correction. Specifically, the NSE score obtained from the model without correction was less than 0, while this score rose up to 0.8316 after applying the model correction procedure, which indicated the necessity of the correction.

**Table 8: Mean score values of all nodes for five measured rainfall events, obtained by using the model with/without correction.**

| | MAE($L \cdot s^{-1}$) | MSE($L^2 \cdot s^{-2}$) | CC | NSE |
|---|---|---|---|---|
| Model without correction | 0.5719 | 4.5045 | 0.1139 | $< 0$ |
| Model with correction | 0.1504 | 0.5919 | 0.9309 | 0.8316 |

To further demonstrate the effect of model correction procedure, we have shown the predicted ponding process at the 6 selected nodes for rainfall event No.5, obtained by using the model with and without correction, as shown in Fig. 15. As shown in the figure, the corrected model performed better at all the selected nodes, e.g., more accurate prediction of start/end time of ponding, more accurate ponding curves (more similar to the measure ones).

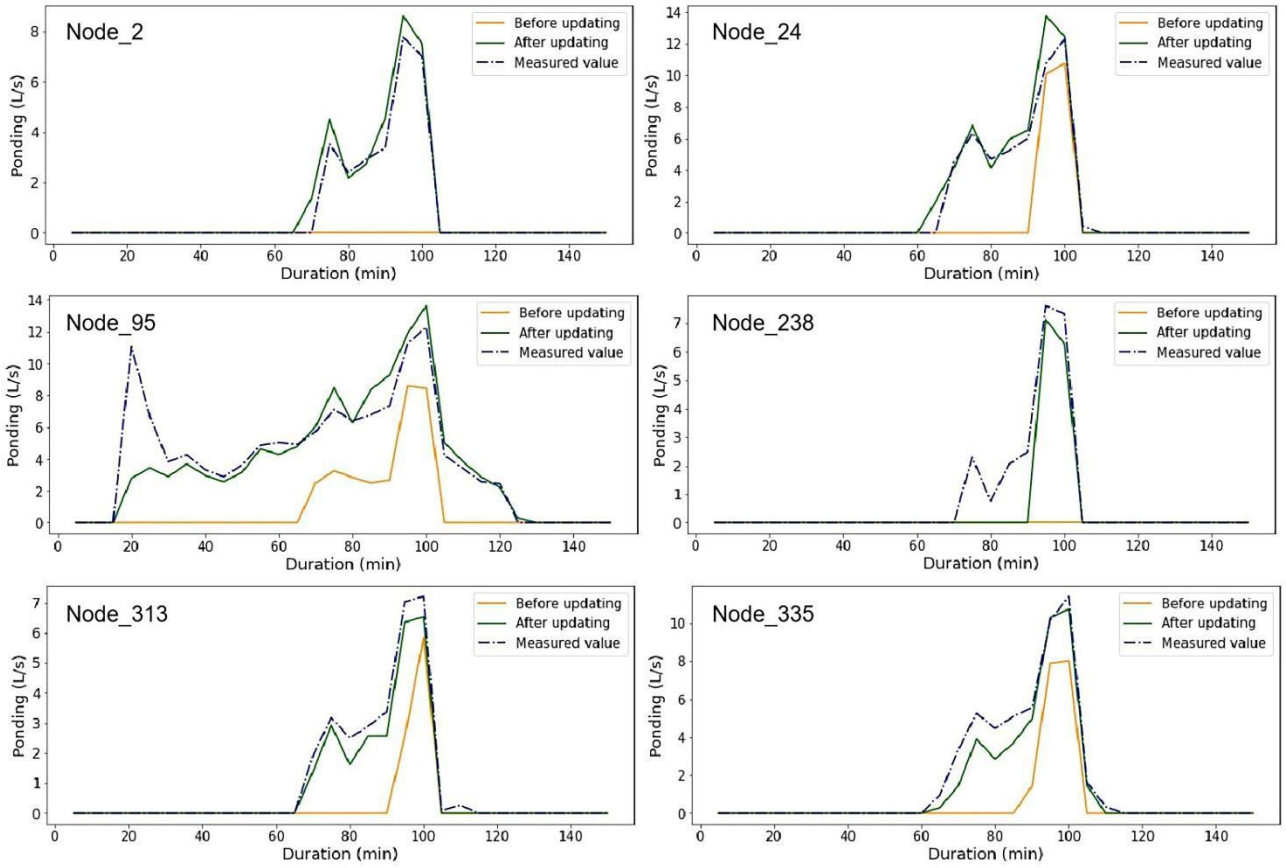



**Figure 15: Comparison between the predicted ponding volume and measured values at the selected nodes in rainfall event No.5.**

All the results shown above demonstrate the superiority of the corrected model compared to the original one, where the monitoring data were introduced in the model correction procedure.

## 4 Discussion

### 4.1 Comparison of neural network structures

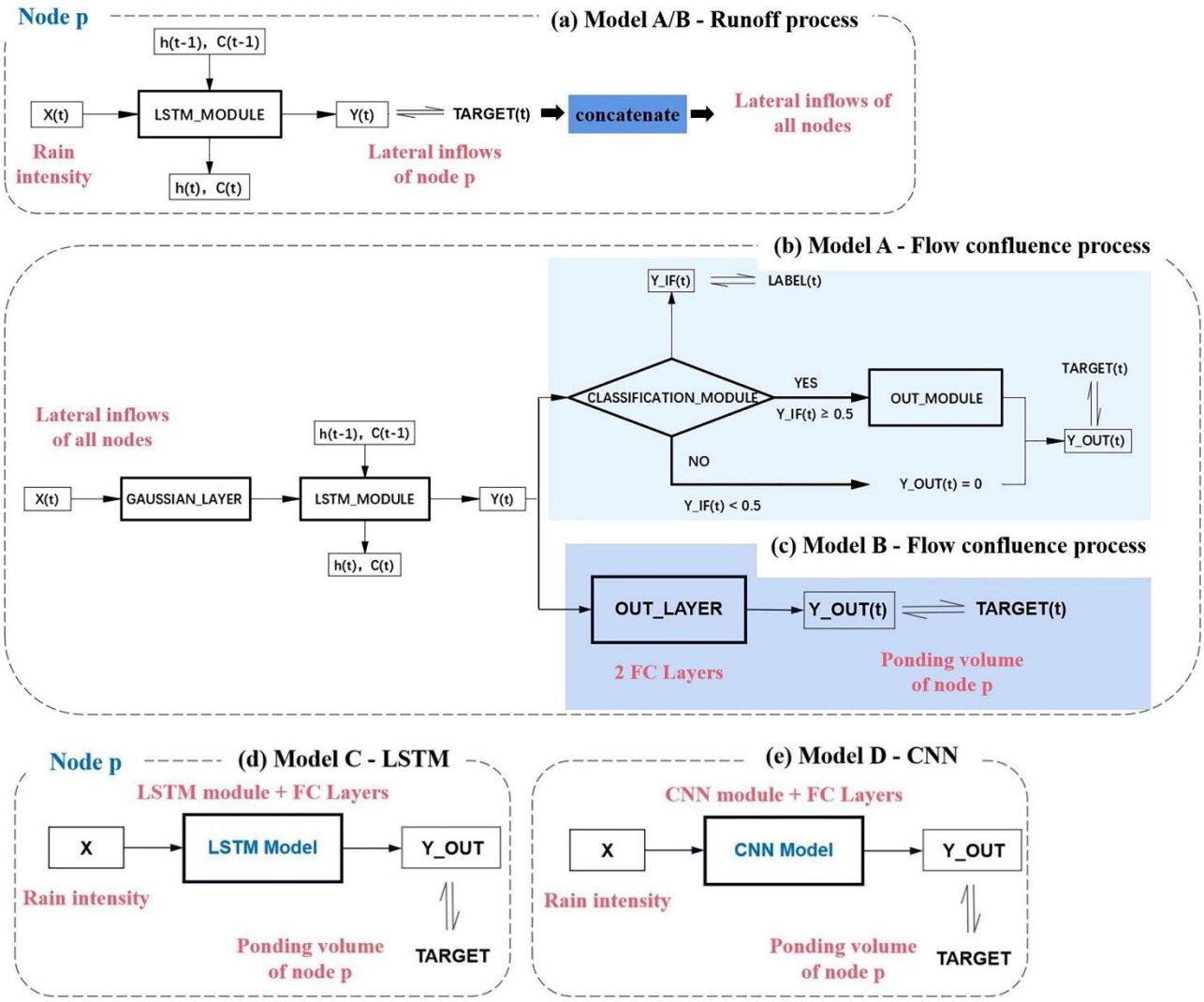
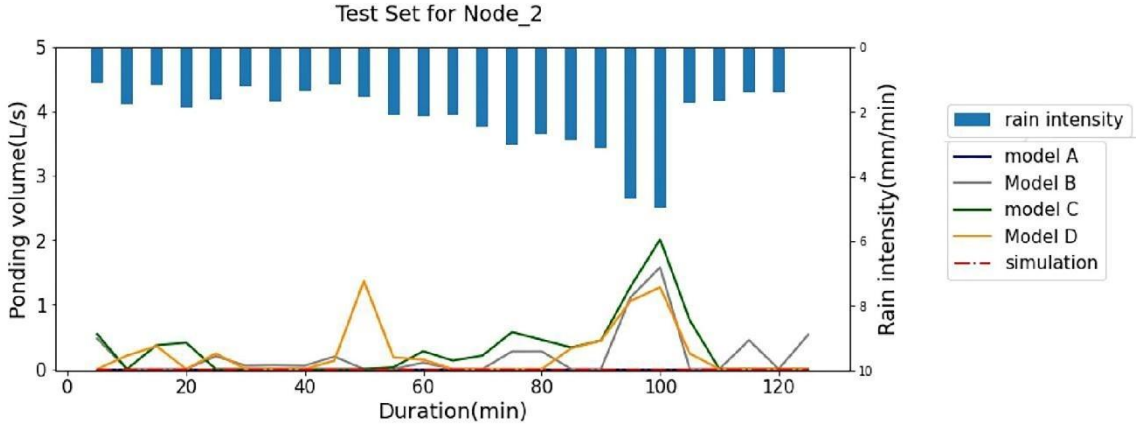


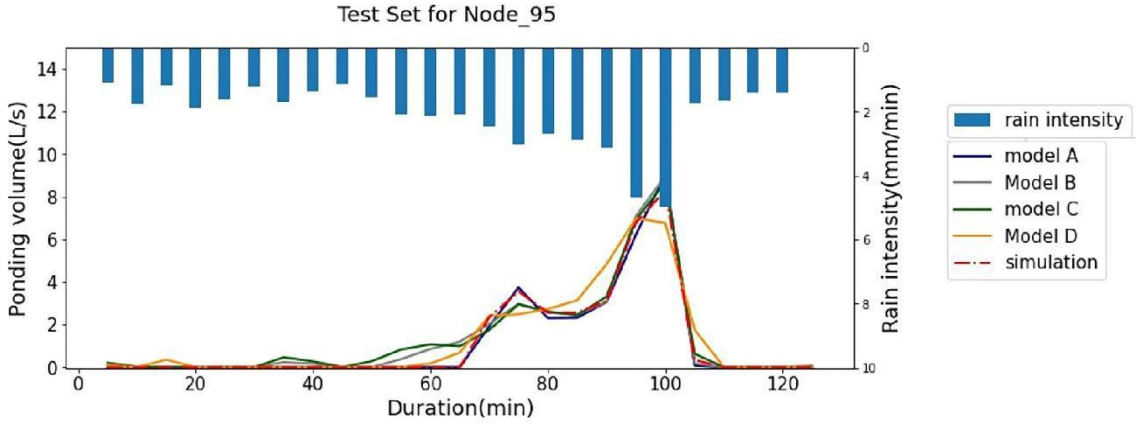

**Figure 16: Schematic diagrams of different network structures for comparison. (a) The same runoff process in model A and B. (b) The multi-target learning in the flow confluence process of model A, marked in light blue. (c) The flow confluence process in model B marked in dark blue. (d) The LSTM structure in model C. (e) The CNN structure in model D.**

The proposed model (termed Model A) was compared with the conventional LSTM structure (termed model B) to show the superiority of the variant of the LSTM structure in the flow confluence process. The schematic diagrams of the two models are shown in Fig. 16 (a ~ c). As shown in the subfigures, Model B has exactly the same structure as model A in the runoff process. The only difference of the two models lies in the flow confluence process, where a multi-task learning mechanism is introduced in the learning process of Model A.

Furthermore, Model A proposed in this paper was compared with two other models (Models C and D) to illustrate the necessity of two processes in tandem, i.e. the runoff and flow confluence processes. The network structures of Models C and D are shown in Fig. 16 (d) and (e), respectively, where the ponding information was obtained directly from rainfall data without extracting the characteristics of lateral inflows.




**(a) Case a where ponding did not occur at Node 2.**




**(b) Case b, where ponding occurred at Node 95.**

**Figure 17: A comparison between the predicted ponding volume by the LSTM-based model (Model A) and by Models B, C, and D for a particular rainfall event.**

Figure 17 shows two examples. In the first example, as shown in the subfigure (a), ponding did not occur at Node 2. However, about 2-4$L \cdot s^{-1}$ ponding volume was falsely reported by the three alternative models (Models B, C, and D), while Model A predicted no ponding at this Node, which was consistent with the simulation (considered as the ground truth). In the second example, as shown in the subfigure (b), where ponding occurred and lasted for about 40 minutes, Model A predicted a more accurate ponding curve than the other three alternative models.

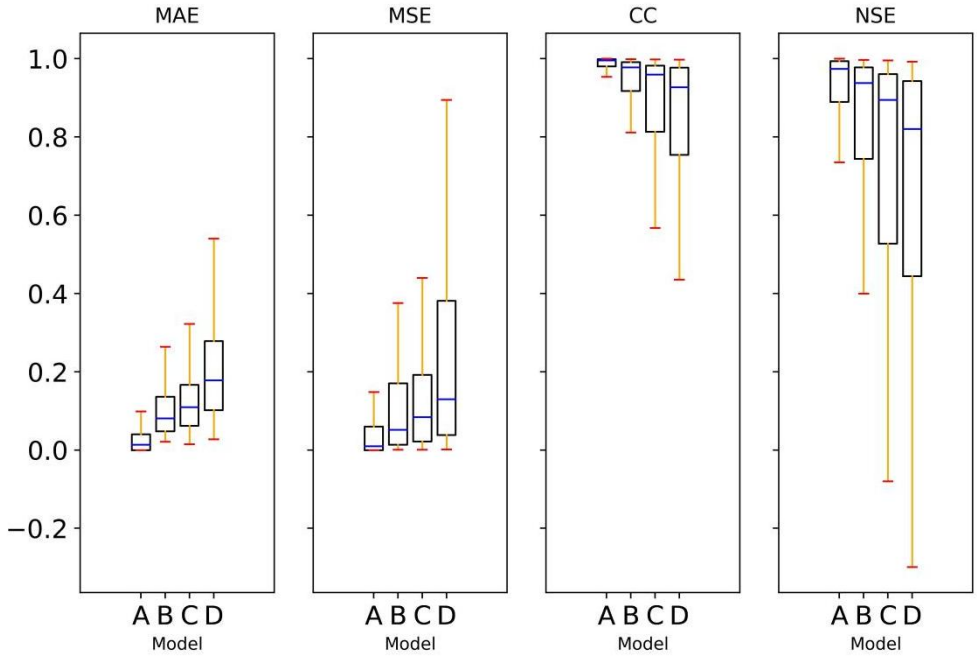



**Figure 18: Comparison of model performance on the ponding volume forecasting. The results of the proposed model A are compared to those obtained from models B, C, and D.**

Figure 18 presents the range of mean score values on the test set for all nodes, obtained by using Models A-D. As shown in the figure, the range of the MAE or MSE score from Model A was half that of model B. The CC scores from model A were very close to 1, while the CC scores from model B varied from about 0.8 to 1. The NSE scores from Model A were generally higher than 0.7, while the NSE scores from model B were unstable and generally lower than those from Model A. Obviously, Model A performed much better than Model B in ponding volume prediction, as indicated by all of the four indicators.

Also as shown in Fig. 18, the obvious superiority of Model A (or B) over Models C and D demonstrates the necessity of having two processes in tandem. Besides, it is also shown in the figure that the range of all these four indicators expanded gradually from Model A to D, which indicated a decreased steadiness.

**Table 9: Mean score values of all nodes obtained from models for predicting the volume of ponding.**

| Model | MAE($L \cdot s^{-1}$) | MSE($L^2 \cdot s^{-2}$) | CC | NSE |
|---|---|---|---|---|
| A | 0.0309 | 0.1624 | 0.9960 | 0.9462 |
| B | 0.0622 | 0.1815 | 0.9578 | 0.8552 |
| C | 0.0849 | 0.2584 | 0.8823 | 0.7424 |
| D | 0.1358 | 0.3480 | 0.9257 | 0.7391 |

Table 9 shows the mean score values at all nodes (on the test set) obtained by using the four models. According to the results, the performance ranking of the four models was Model A > Model B > Model C > Model D.

The comparative analyses above indicated that the LSTM-based model proposed in this paper had remarkable superiority over the other three alternatives in ponding volume forecasting. There are two reasons behind this. First, the proposed model had two processes in tandem: the runoff and flow confluence processes. The second is due to the auxiliary classification task introduced in the flow confluence process. The two tandem processes reduced the computational burden of this data-driven approach and avoided interference with each other during training. While the classification task introduced facilitated the capability of the model to identify ponding.

### 4.2 The influence of the number of monitoring points on model correction

It was easy to spot from the trial that the performance of the corrected model depended on whether the layout of the monitoring points could reflect the hydraulic conditions of the pipe network. An unreasonable design of the monitoring equipment might lead to a failure in model correction.

There were 15 level gauges and three flowmeters in the case pipe network, as shown in Fig. 7(b). To analyse how the number of monitoring points impacted the performance of the revised model, different numbers of monitoring points were randomly selected as a quantitative control group. Figure 19 presented the evaluation results of the revised model on ponding volume forecasting obtained by using different numbers of monitoring points.

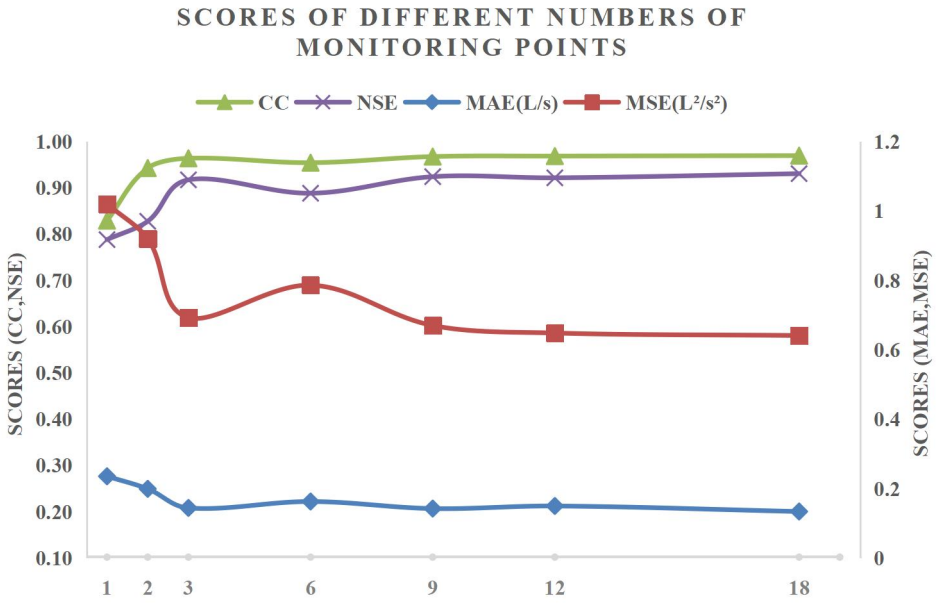



**Figure 19: Score values obtained by using different numbers of monitoring points in the model correction process.**

As shown in the figure, the NSE scores stayed around 0.9 when the number of monitoring points exceeded 6, the CC scores showed similar trends as the NSE scores. Besides, the other scores showed the opposite trends. It turned out that, when the number of monitoring points was over 1 per hectare, increasing the number of monitoring points further had limited effect on

improving the accuracy of the corrected model. However, when the number of monitoring points was below 0.5 per hectare (i.e., the number of monitoring points was less than 3), it is highly effective to increase the number of monitoring points in the pipe network. For example, the NSE score was lower than 0.8 when the number of monitoring points was only 1.

In summary, one monitoring point per hectare is the critical point. If the number of monitoring points was less than this limit, the performance of the revised model could not be guaranteed.

## 5 Conclusions

This work aims at promoting the application of deep learning in urban flood forecasting. Specifically, we have proposed an optimized LSTM-based approach in this study, which can quickly identify and locate ponding with relatively high accuracy. According to the research results, the main conclusions of this study are summarized as follows:

1 The proposed model is constructed by two tandem processes (runoff process and flow confluence process) and utilizes a multi-task learning mechanism to achieve high accuracy. Over 15000 designed rainfall events were used for model training, which covers various extreme weather conditions. The median score of NSE for ponding forecasting is greater than 0.95, and the mean accuracy at any node to determine whether ponding occurs reaches higher than 0.98.

2 The superiority of the proposed model has been demonstrated by comparing with two widely used deep learning models: (traditional) LSTM and CNN models.

The superiority of the proposed model having two tandem processes is proved by comparing with LSTM and CNN structures with a single process. The mean NSE score for ponding volume forecasting of the proposed model is 0.9462, while that of LSTM and CNN structures with a single process is 0.7424 and 0.7391 respectively. Then, the superiority of the proposed model with a LSTM variant is demonstrated by a comparison with the conventional LSTM structure also with two tandem processes. As shown in Table 9, the mean NSE score of the latter is 0.8552.

3 An approach to model modification using real-life monitoring level and flow data is proposed in this paper. The proposed LSTM-based model is further calibrated to achieve better accuracy.

The LSTM-based model is corrected using two steps. First, the runoff process is corrected with the measured rain, level, and flow data referring to Parameter-based (Model-based) Transfer Learning. Then, the flow confluence process is updated using the updated lateral inflows at all nodes and the measured ponding volume. As shown in Table 8, the mean CC score at all nodes of the model with correction is 0.9309, while that of the model without correction is 0.1139.

Overall, the proposed LSTM-based approach provides a new possibility for early warning and forecasting of ponding in urban drainage system. In this study, all operations were conducted in an offline mode. In the future study, we will explore the capability of the proposed model in real-time event analysis. Furthermore, we will optimize the model by considering the influence of two-dimensional overland flow in ponding volume prediction.

*Code availability.* The pieces of code used for all analyses are available from the authors upon request.

*Data availability.* All data used in this study are available from the authors upon request.

*Competing interests.* The contact author has declared that neither they nor their co-authors have competing interests.

*Acknowledgment.* The authors gratefully acknowledge the financial supports by the National Natural Science Foundation of China under Grant numbers 51978493, and also thank all the team members for their insightful comments and constructive suggestions to polish this paper in high quality.

**References**

Abou Rjeily, Y., Abbas, O., Sadek, M., Shahrour, I. and Hage Chehade, F.: Flood forecasting within urban drainage systems using NARX neural network, Water Science and Technology, 76, 2401-2412, https://doi.org/10.2166/wst.2017.409, 2017.

Archetti, R., Bolognesi, A., Casadio, A. and Maglionico, M.: Development of flood probability charts for urban drainage network in coastal areas through a simplified joint assessment approach, Hydrology and Earth System Sciences, 15, 3115-3122, http://dx.doi.org/10.5194/hess-15-3115-2011, 2011.

Aryal, D. et al.: A Model-Based Flood Hazard Mapping on the Southern Slope of Himalaya, Water, 12, 540, https://doi.org/10.3390/w12020540, 2020.

Bai, Y., Bezak, N., Sapač, K., Klun, M. and Zhang, J.: Short-Term Streamflow Forecasting Using the Feature-Enhanced Regression Model, Water Resources Management, 33, 4783-4797, https://doi.org/10.1007/s11269-019-02399-1, 2019.

Balstrøm, T. and Crawford, D.: Arc-Malstrøm: A 1D hydrologic screening method for stormwater assessments based on geometric networks, Computers & Geosciences, 116, 64-73, https://doi.org/10.1016/j.cageo.2018.04.010, 2018.

Bergstra, J., Yamins, D., Cox, D.D., 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures.

Cai, B. and Yu, Y.: Flood forecasting in urban reservoir using hybrid recurrent neural network, Urban Climate, 42, 101086, https://doi.org/10.1016/j.uclim.2022.101086, 2022.

Chiang, Y., Li-Chiu, C., Meng-Jung, T., Yi-Fung, W. and C., F.: Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites, Hydrology and Earth System Sciences, 14, 1309-1319, https://doi.org/10.5194/hess-14-1309-2010, 2010.

Djordjević, S., Prodanović, D. and Maksimović, Č.: An approach to simulation of dual drainage, Water Science and Technology, 39, 95-103, https://doi.org/10.1016/S0273-1223(99)00221-8, 1999.

Djordjević, S., Prodanović, D., Maksimović, Č., Ivetić, M. and Savić, D.: SIPSON – Simulation of Interaction between Pipe flow and Surface Overland flow in Networks, Water Science and Technology, 52, 275-283, https://doi.org/10.2166/wst.2005.0143, 2005.

Guo, K., Guan, M. and Yu, D.: Urban surface water flood modelling-a comprehensive review of current models and future challenges, Hydrology and Earth System Sciences, 25, 2843-2860, http://dx.doi.org/10.5194/hess-25-2843-2021, 2021.

Hossain Anni, A., Cohen, S. and Praskievicz, S.: Sensitivity of urban flood simulations to stormwater infrastructure and soil infiltration, Journal of Hydrology, 588, 125028, https://doi.org/10.1016/j.jhydrol.2020.125028, 2020.

Huong, H.T.L. and Pathirana, A.: Urbanization and climate change impacts on future urban flooding in Can Tho city, Vietnam, Hydrology and Earth System Sciences, 17, 379-394, http://dx.doi.org/10.5194/hess-17-379-2013, 2013.

Jamali, B. et al.: A rapid urban flood inundation and damage assessment model, Journal of Hydrology, 564, 1085-1098, https://doi.org/10.1016/j.jhydrol.2018.07.064, 2018.

Kao, I., Zhou, Y., Chang, L. and Chang, F.: Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting, Journal of Hydrology, 583, 124631, https://doi.org/10.1016/j.jhydrol.2020.124631, 2020.

Kratzert, F. et al.: Toward Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning, Water Resources Research, 55, https://doi.org/10.1029/2019WR026065, 2019.

Kratzert, F. et al.: Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets, Hydrology and Earth System Sciences, 23, 5089-5110, https://doi.org/10.5194/hess-23-5089-2019, 2019.

Kuczera, G. et al.: Joint probability and design storms at the crossroads, Australian journal of water resources, 10, 63-79, https://doi.org/10.1080/13241583.2006.11465282, 2006.

Leandro, J. and Martins, R.: A methodology for linking 2D overland flow models with the sewer network model SWMM 5.1 based on dynamic link libraries, Water Science and Technology, 73, 3017-3026, https://doi.org/10.2166/wst.2016.171, 2016.

Moy De Vitry, M., Kramer, S., Dirk Wegner, J. and Leitao, J.P.: Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network, Hydrology and Earth System Sciences, 23, 4621-4634, http://dx.doi.org/10.5194/hess-23-4621-2019, 2019.

Mudashiru, R.B., Sabtu, N., Abustan, I. and Balogun, W.: Flood hazard mapping methods: A review, Journal of hydrology (Amsterdam), 603, 126846, https://doi.org/10.1016/j.jhydrol.2021.126846, 2021.

Pan, S.J. and Yang, Q.: A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, 22, 1345-1359, https://doi.org/10.1109/TKDE.2009.191, 2010.

Pilgrim, D.H. and Cordery, I.: Rainfall Temporal Patterns for Design Floods, Journal of the Hydraulics Division, 101, 81-95, https://doi.org/10.1061/JYCEAJ.0004197, 1975.

Rahman, A., Weinmann, P.E., Hoang, T.M.T. and Laurenson, E.M.: Monte Carlo simulation of flood frequency curves from rainfall, Journal of hydrology, https://doi.org/10.1016/S0022-1694(01)00533-9, 2002.

Skougaard Kaspersen, P., Hoegh Ravn, N., Arnbjerg-Nielsen, K., Madsen, H. and Drews, M.: Comparison of the impacts of urban development and climate change on exposing European cities to pluvial flooding, Hydrology and Earth System Sciences, 21, 4131-4147, http://dx.doi.org/10.5194/hess-21-4131-2017, 2017.

Yang, T., Hwang, G., Tsai, C. and Ho, J.: Using rainfall thresholds and ensemble precipitation forecasts to issue and improve urban inundation alerts, Hydrology and Earth System Sciences, 20, 4731-4745, http://dx.doi.org/10.5194/hess-20-4731-2016, 2016.