



# High temporal resolution rainfall–runoff modeling using long-short-term-memory (LSTM) networks

Wei Li<sup>1</sup> · Amin Kiaghadi<sup>1</sup> · Clint Dawson<sup>1</sup>

Received: 16 January 2020 / Accepted: 2 May 2020 / Published online: 5 June 2020  
© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

Accurate and efficient models for rainfall–runoff (RR) simulations are crucial for flood risk management. Most rainfall models in use today are process-driven; i.e., they solve either simplified empirical formulas or some variation of the St. Venant (shallow water) equations. With the development of machine-learning techniques, we may now be able to emulate rainfall models using, for example, neural networks. In this study, a data-driven RR model using a sequence-to-sequence long-short-term-memory (LSTM) network was constructed. The model was tested for a watershed in Houston, TX, known for severe flood events. The LSTM network's capability in learning long-term dependencies between the input and output of the network allowed modeling RR with high resolution in time (15 min). Using 10-year precipitation from 153 rainfall gages and river channel discharge data (more than 5.3 million data points), and by designing several numerical tests, the developed model performance in predicting river discharge was tested. The model results were also compared with the output of a process-driven model gridded surface subsurface hydrologic analysis (GSSHA). Moreover, physical consistency of the LSTM model was explored. The model results showed that the LSTM model was able to efficiently predict discharge and achieve good model performance. When compared to GSSHA, the data-driven model was more efficient and robust in terms of prediction and calibration. Interestingly, the performance of the LSTM model improved (test Nash–Sutcliffe model efficiency from 0.666 to 0.942) when a selected subset of rainfall gages based on the model performance, were used as input instead of all rainfall gages.

**Keywords** Hydrologic analysis · GSSHA · Recurrent neural network · Machine learning

## 1 Introduction

Flooding is considered the leading cause of natural-disaster losses in the USA with an average annual damage of \$7.95 billion (1984–2013, adjusted to 2014 inflation) [37]. Implementing flood management strategies without a reliable predictive rainfall–runoff (RR) modeling framework is not possible. RR modeling, which aims at predicting the streamflow hydrograph from precipitation input, is

intensively studied and used to support flood assessment [13, 35, 38, 45, 52, 53]. In addition, RR models are required to provide reliable discharge input for storm surge models when real-time data are not available due to the absence of measuring gages. Such coupling is very important to simulate the cascading effects of storm surge, local runoff, and compound flooding in coastal areas.

RR models can be categorized as process-driven and data-driven models [25, 33, 51]. While process-driven methods are composed of analytical and empirical formulae based on physical phenomena, data-driven models rely on interpolating and extrapolating data. During the past two decades, multiple process-driven hydrologic models such as Interconnected Channel and Pond Routing Model (ICPR), Hydrologic Engineering Center's River Analysis System (HEC-RAS), and Gridded Surface Subsurface Hydrologic Analysis (GSSHA) for RR simulation have been developed [2, 7, 11, 23].

✉ Amin Kiaghadi  
kiaghadi@utexas.edu

Wei Li  
wei@ices.utexas.edu

Clint Dawson  
clint@oden.utexas.edu

<sup>1</sup> Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin 78712, TX, USA

Although much progress has been made recently, even state of the art process-driven models like GSSHA rely on accurate meteorological input data that are changing due to human/natural activities, which adds difficulty to constructing a production-level model incipiently. Furthermore, accurate prediction of RR requires extensive calibration of the multiphysics model that is computationally expensive and requires intensive data availability and entry. In addition, using process-driven models make it more difficult to build a coupled coast flood prediction model that considers the coupled interactions of hurricane storm surge and associated RR [47]. During a flood event, channel discharge information computed by RR model will be passed to a surge model as a flux boundary condition. Concurrently water surface elevation computed by the surge model will be enforced on the RR watershed boundary as a Dirichlet boundary condition. The overhead caused by this message passing between RR and storm surge model will affect the overall computational efficiency.

Recently using deep neural networks for real time flood prediction has been made possible by the increasing amount of collected hydrologic data. Antithetical to process-driven models, data-driven models such as artificial neural network (ANN) that have been widely applied to streamflow prediction, e.g., [9, 20, 22, 25, 42, 43, 50], are more robust to meteorological data changes. This robustness is due to the nature of their training data, batched learning, and relatively inexpensive calibration process. Due to its capability of modeling highly nonlinear relationships between input and output, the ANN model has generated promising results for RR simulation.

When it comes to time series data, standard feed forward neural network has its limitations. Feed forward neural networks are designed based on the assumption that the training and test examples (data points) are independent. Thus, the entire state of the network is erased after processing each example [48]. This assumption is not desired when data points are inherently related. Moreover, to deal with time series data, a standard ANN model (feed-forward) would require choosing a fixed-sized sliding window over the dataset. Tuning the size of this sliding windows for the best predictive accuracy adds extra work to the model selection [22]. This limitation becomes more significant in flood assessment with finer time resolution (i.e., 15 min). In this case, long-term dependencies prevail due to the small time step size and cannot be learned by ANN because they are not captured within the fixed-sized time windows.

More recently, a class of ANNs known as recurrent neural network (RNN), a deep learning algorithm, has attracted much attention and shown success in solving sequential problems such as machine translation, speech recognition, and handwriting recognition [14]. Even though

the idea of RNN was proposed in the 1980s [36], the applications of RNN in hydrologic engineering are relatively more recent [24, 46]. RNNs are networks with loops in them, allowing information to persist. RNN can exploit the sequential pattern in the data while preserving feed-forward NN's ability to model nonlinear relationship between input and output via cycles formed by the hidden nodes in the network [32]. A standard RNN has very simple looping units, such as a single layer with hyperbolic tangent (tanh) activation. To cope with the vanishing gradient challenge [21] for standard RNN and learn longer-term dependencies in sequential data, long short-term memory (LSTM)-based RNN systems have been developed [18]. LSTM's success has encouraged groups to explore its capability in time series forecasting of river discharge and other applications [26, 28, 29, 34, 41, 49, 51].

All of the aforementioned LSTM models have been using both rainfall and flow at previous time steps to predict future flow. Even though the prediction uncertainty associated with time series forecast using LSTM is not analytically available yet, studies [34, 49, 51] have shown increasing error in predicting flow by the passage of time. Despite some examples in using RNN for hydrologic modeling [15], the literature obviously lacks an LSTM model that predicts future river flow purely based on precipitation input to address this accumulative uncertainty problem. In addition, such a model, capable of simulating longer events only using precipitation data as input, is more desirable for flood management applications and dynamic coupling with surge models.

Ubiquitous as deep learning systems are, they are often criticized for their lack of interpretability. There are multiple motivations to interpret deep learning models [10], among which two aspects raise the most concerns: (1) How can the prediction be trusted when the model is not interpretable? (2) How to select input features (rainfall gages) for the model? These two questions are especially difficult to answer for LSTM models. Unlike some machine learning models that have clearly defined importance metrics such as the random forest algorithm [6], there is no simple way to define such a metric and offer insights to the learned LSTM model. Besides, the complicated structure of LSTM unit adds more difficulty to understand the prediction. While the importance of the first question is obvious, the second question is equally important as it is found that scrutinized gage selection can improve the model predictions [1, 31]. Thus, it is critical to understand the data-driven models and justify their results based on the physical intuition in addition to making good predictions.

In this paper, an LSTM network was applied to build a data-driven model for streamflow prediction in a urban watershed on a 15-min scale and compared it with a benchmark process-driven model (GSSHA). The objectives

of this paper are (1) to build a data-driven model for streamflow prediction using precipitation as the only input in an urban watershed by applying LSTM network, (2) to compare the prediction accuracy and efficiency of the developed model with a benchmark process-driven model (GSSHA) and observational data, (3) to evaluate to what extent the model results can be justified based on the physical characteristics of the modeled watershed, and (4) to propose a fast methodology to reduce the dimension of input data to the model through an efficient feature selection approach.

## 2 Study area and data acquisition

Figure 1 shows the location of the Brays Bayou watershed, Brays Bayou and its tributaries located in southwest of Harris County and northeast of Fort Bend County, Texas was selected for this study. Brays Bayou drains freshwater from 329 square kilometers of a heavily urbanized and populated watershed and discharges into the Houston Ship

Channel [5]. Brays Bayou has had a history of floods; just in the last 18 years Tropical Storm Allison (2001), Hurricane Ike (2008), the Memorial Day Flood (2015), the Tax Day Flood (2016), and Hurricane Harvey (2017) caused significant flooding and billions of dollars of property damage [37].

15-min precipitation data from 2007 to 2017 were compiled from 153 rainfall gages maintained by the Harris County Flood Control District (HCFCD) and 15 min flow data were obtained from the United States Geological Survey (USGS) gages [19]. Within the Brays Bayou watershed, there are 16 rainfall gages and five flow gages. In this study, only one freshwater gage located very close to the watershed outlet (see gage 08075000 in Fig. 1) was used to compile flow data for the purpose of training, calibration and validation.

Land elevation was extracted from the 10 m resolution US National Elevation Dataset (NED) in WMS and assigned to the grid. The 15-class land use data was compiled from 30 m resolution US National Land Cover Database (NLCD).

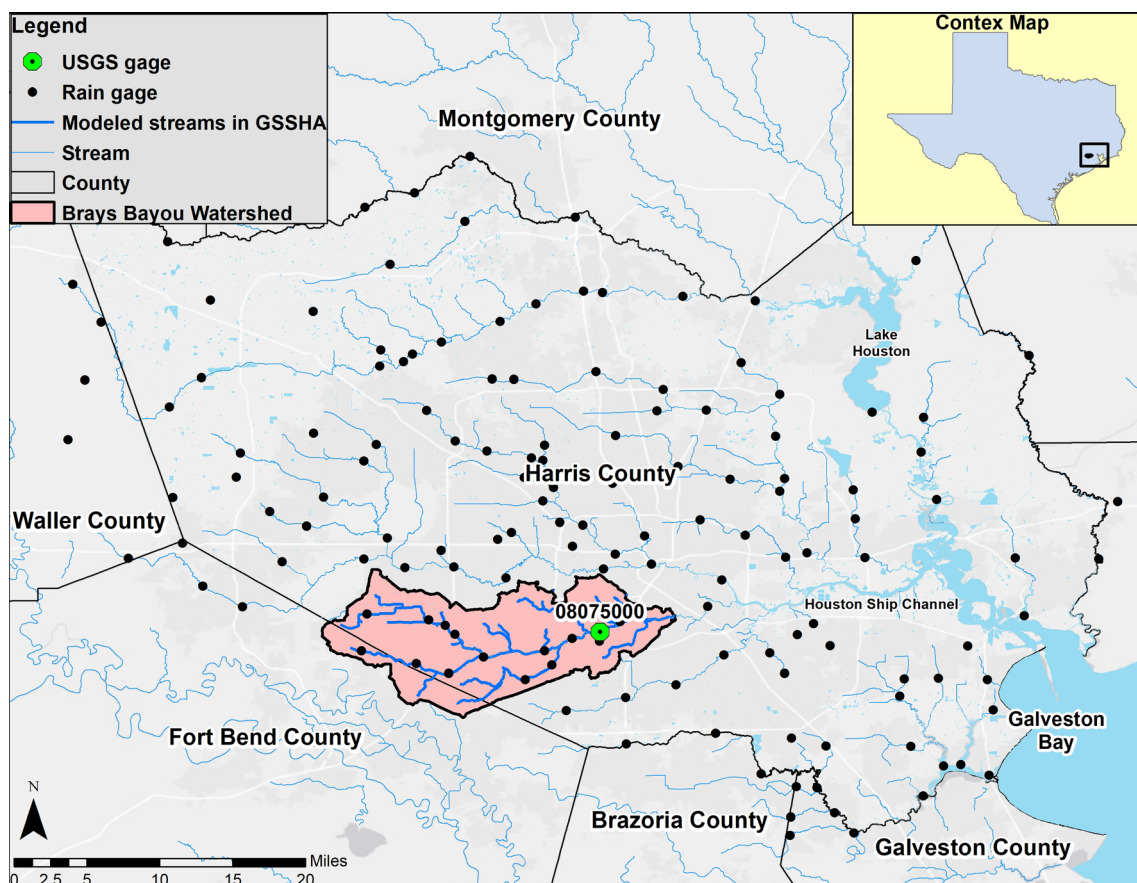


Fig. 1 Study area

### 3 Methods

#### 3.1 Gridded surface subsurface hydrologic analysis (GSSHA) model setup

GSSHA is developed and actively operated by the Engineer Research and Development Center (ERDC) of the United States Army Corps of Engineers (USACE). GSSHA is an open source distributed-parameter hydrologic model capable of coupling multiple physical interactions among 1D channel flow, 2D overland flow, infiltration and groundwater flow, precipitation interception, snow melting, and evapotranspiration [12]. Among the process-driven models, GSSHA has been widely used by many researchers for various purposes from total maximum daily loads (TMDLs) to compound flooding; in the period of 2000–2017, GSSHA has been used in more than 85 scientific/technical projects [4]. More recently, [40] coupled GSSHA with the state-of-the-art surge modeling system (ADCIRC-SWAN) to simulate compound flooding on the east coast of Puerto Rico. Other recent studies used GSSHA to improve the parameterization of the Storm Water Management Model [16], and to evaluate the performance of satellite-based precipitation products in comparison to radar data [17]. Thus, it is chosen as the benchmark model for this study. Considering the location, geography, and objectives of the study only surface flow routing processes were activated.

In this paper, a GSSHA model was built for the study area using the Watershed Modeling System (WMS) version 10.1. WMS is a watershed RR simulation and modeling software application from Aquaveo™ [8]. The software supports a number of hydraulic and hydrologic models including GSSHA that can be used to create drainage basin simulations. A uniform 2-D grid with 56,606 cells with a dimension of 100 by 100 m for 2D overland flow was constructed. The streams were represented by 49 reaches of trapezoidal channels. The channel nodes have an average length of 470 m in the longitudinal direction. The cross-section geometry is approximated based on an existing HEC-RAS model for Brays Bayou developed by the USACE. To compute 2D overland flow, the alternating direction explicit (ADE) method was chosen in GSSHA. To assign surface roughness parameters (Manning coefficient) an index map was created using 15-class land use data. For each land use class [see Fig. 8 in “Appendix”], Manning coefficient recommended by the National Resource Conservation Service (NRCS) was used (see Table 4 in “Appendix”). The channel flow is modeled using explicit diffusive wave method. The precipitation data were obtained from 16 rainfall gages inside the Brays Bayou watershed. The distributed rainfall was then

interpolated using Theissen polygon and inverse distance weighted methods.

Limited by the computational expense, the GSSHA model used in this study was calibrated on a period from 2014/12/13 to 2015/01/30, which consists of a series of rainfall events. The highest peak flow of the events is 317 cm. River channel’s Manning’s coefficient was set as the only calibration factor and restricted within the range between 0.001 and 0.02. The optimal Manning’s coefficient is found to be 0.003 by grid search within this range. The RMSE of the calibrated model is 14.03 and NSE of is 0.757.

Furthermore, four flood events in 2017 with different scales were simulated using the calibrated model to be compared with the LSTM model’s result and observed data. To be consistent with the data-driven model, discharge at the chosen USGS fresh water gage (08075000) is set as the observation point.

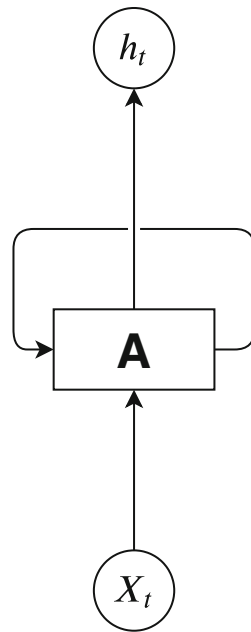
#### 3.2 Long-short-term-memory (LSTM) network

In this study, a standard LSTM network was used to predict the discharge from rainfall data. The LSTM network is an RNN composed of LSTM units. RNN structures have been explained elsewhere [30], but in brief and as shown in Fig. 2a, at each time step  $t$ , a neural network,  $A$ , looks at some input  $x_t \in \mathbb{R}^d$ , where  $d$  is the dimension of the input, and hidden state from the last time step  $h_{t-1}$ , and outputs a value  $h_t \in \mathbb{R}$ . A loop allows information to be passed from one step of the network to the next. At the next time step  $t + 1$ , the new input  $x_{t+1}$  and hidden state  $h_t$  are fed into the network, and new hidden state  $h_{t+1}$  is computed. In theory, RNNs are capable of handling “long-term dependencies”. For instance, initial input  $X_0$  could affect the hidden state value 500 steps later ( $h_{500}$ ). Unfortunately, in practice, due to numerical limitation during the optimization stage, RNNs consisting of single layer of artificial neurons are unable to learn to connect the long-term dependency [3]. Instead of an artificial neuron, an LSTM unit contains a memory cell  $g_t$  and three gates. These gates are input gate  $i_t$ , forget gate  $f_t$ , and output gate  $o_t$  [18].

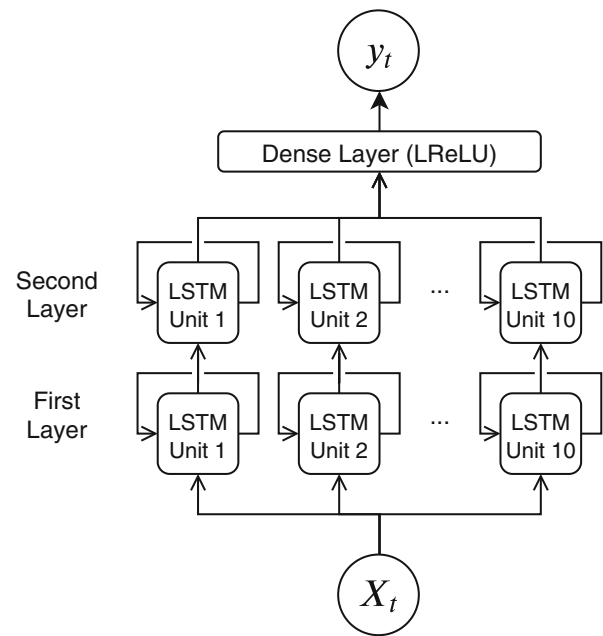
At time step  $t$ , the LSTM unit takes input  $x_t$ , hidden states  $h_{t-1}$ . Then, it updates the hidden states following (1):

$$\begin{aligned} i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\ f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\ g_t &= \sigma(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\ o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\ c_t &= f_t c_{t-1} + i_t g_t \\ h_t &= o_t \tanh(c_t) \end{aligned} \quad (1)$$

**Fig. 2** Architecture for recurrent network and the developed LSTM network



(a) Structure of RNN.



(b) Developed two layer LSTM network with 10 hidden units.

where  $\sigma$  is the sigmoid function;  $W_{ii}, W_{if}, W_{ig}$ , and  $W_{io}$  are the input-hidden weights;  $W_{hi}, W_{hf}, W_{hg}$ , and  $W_{ho}$  are the hidden-hidden weights;  $b_{ii}, b_{if}, b_{ig}$ , and  $b_{io}$  are the input-hidden biases;  $b_{hi}, b_{hf}, b_{hg}$ , and  $b_{ho}$  are the hidden-hidden biases.

Depending on the size of the watershed, the peak of generated runoff can be observed from a couple of hours to a couple of weeks after the event. For instance, for a time step of 15 min, a rainfall event that lasted for a week would have more than 650 steps in time. To accurately model the RR process, the model is required to memorize the effect of precipitation from the beginning of the event, which is numerically difficult for standard RNN. Thus, such a structure shown in (1) is proposed to cope with the vanishing gradient problems that can be encountered when training standard RNNs [18]. With the objective of channel discharge prediction, a two-layer LSTM network with 10 hidden units in each layer was designed as shown in Fig. 2b. A standard LSTM implementation from the open source deep learning platform PyTorch [39] was adopted in this study to develop the data-driven model.

In this study, input  $x_t = (x_{t_1}, x_{t_2}, \dots, x_{t_n})$  is the vector of precipitation gage readings at time  $t$ .  $x_{t_i}$  corresponds to the reading of the  $i$ -th precipitation gage at time  $t$ . Note that flow at any of the previous time steps ( $y_{t-k}$ ) is not within the input vector indicating the developed model is not a time series model. In other words, it does not depend on the immediate past observation to make prediction. For each time step  $t$ , the precipitation readings are input into the

network, and  $h_t \in \mathbb{R}^p$  is computed by the two layers of LSTM network  $\mathcal{M}_{LSTM}: h_t, c_t = \mathcal{M}_{LSTM}(x_t, h_{t-1}, c_{t-1})$ . Each hidden unit works independently. Formally, the LSTM network  $\mathcal{M}_{LSTM} : \mathbb{R}^{2p} \mapsto \mathbb{R}^{2p}$  is defined as:

$$\begin{aligned} h_l &= o_l \tanh(c_l) \\ c_l &= f_l c_{l-1} + i_l g_l \end{aligned} \tag{2}$$

where dimension of  $h_t$  and  $c_t$ ,  $p = 10$  is the number of hidden units set by user.  $h_l, c_l$  are the  $l$ th element of the  $h_t, c_t$  vector. The gates and cell state  $o_l, f_l, i_l, g_l$  are updated following (1). Since the output dimension is 1 (we are only trying to predict the outlet discharge), another fully connected layer  $\mathcal{M}_{fc} : \mathbb{R}^p \mapsto \mathbb{R}$  would transform the  $h_t$  to the output  $\hat{y}_t: \hat{y}_t = \mathcal{M}_{fc}(h_t)$ :

$$\mathcal{M}_{fc}(h_t) = g_a \left( \beta_0 + \sum_{l=1}^p \beta_l h_l \right) \tag{3}$$

where  $g_a$  is the activation function; in the context of artificial neural networks, the activation function of a node defines the output of that node for a given input or set of inputs. A nonlinear activation function allows the neural networks to model nonlinear relationship between the input and output. This function is also known as the transfer function [30]. An activation function defined as the positive part of its argument is called rectifier:

$$g_a(x) = x^+ = \max(0, x), \tag{4}$$

where  $x$  is the input to a neuron. Leaky rectified linear unit (LReLU) is a generalization of rectifier where a small,

positive gradient is allowed when the unit is not *active* (input to the neuron is not positive):

$$g_a(x) = \begin{cases} x & \text{if } x > 0, \\ 0.01x & \text{otherwise.} \end{cases} \quad (5)$$

As shown in (1) and (2), an initial hidden state is required for the LSTM network to start a forward propagation, i.e.,  $h_1, c_1 = \mathcal{M}_{LSTM}(x_1, h_0, c_0)$ . After some preliminary test runs, a modification was made to the model to set the initial hidden states  $(h_0, c_0)$  as learnable parameters instead of randomly initializing it to improve the prediction performance at the initial stage. The initial hidden states  $(h_0, c_0)$  are fixed after training. This change was justified because hydrologic models are generally set up with an initial condition (base flow) which exists before the rainfall event.

Like any supervised learning algorithm, calibrating the LSTM model requires *training* the model by optimizing the objective function  $\mathcal{L}(\hat{y}_t, y; w)$  under some constraint  $C$ , where  $\hat{y}_t$  and  $y$  are the prediction and observation, respectively.  $w$  is the learnable parameter:

$$\begin{aligned} & \arg \min_w \\ & \text{subject to} \end{aligned} \quad (6)$$

In this study, the objective function is defined as the mean square error (MSE) between the prediction and observation:

$$\mathcal{L}(\hat{y}_t, y; w) = \frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y)^2. \quad (7)$$

To avoid over-fitting of the deep learning models, regularization by adding penalty to the learnable parameters is a widely used approach [30]. In practice, regularization  $\mathcal{R}(w)$  is added to the objective function. A regularized version of the optimization problem becomes,

$$\begin{aligned} & \arg \min_w \\ & \text{subject to} \end{aligned} \quad (8)$$

where  $C$  is the constraint on  $w$ .

In this work,  $l_2$  regularization is used to prevent the model from over-fitting to the training data,

$$\mathcal{R}(w) = \lambda \|w\|_2^2 / 2. \quad (9)$$

where  $\lambda$  is the regularization parameter ( $10^{-6}$ ). Larger  $\lambda$  corresponds to more regularization. Note that this technique is also known as weight decay because when applying standard stochastic gradient descent (SGD), it is equivalent to updating the weight in this way:

$$w_{i+1} = w_i - \lambda w_i - \alpha \frac{\delta L}{\delta w} |_{w_i} \quad (10)$$

where  $w_i$  is the learnable parameters at step  $i$ ,  $\alpha$  is the learning rate, and  $\frac{\delta L}{\delta w} |_{w_i}$  is the stochastic gradient approximation at step  $i$ . Thus, at each step, the weight  $w$  decays by  $(1 - \lambda)$ . For standard stochastic gradient descent (SGD), weight decay can be made equivalent to  $l_2$  by a reparameterization of the weight decay factor based on the learning rate. However, this is not the case for adaptive gradient descent methods including Adam optimization [27], which is used in this study with a learning rate of  $10^{-4}$ .

### 3.3 LSTM model training, validation, and evaluation

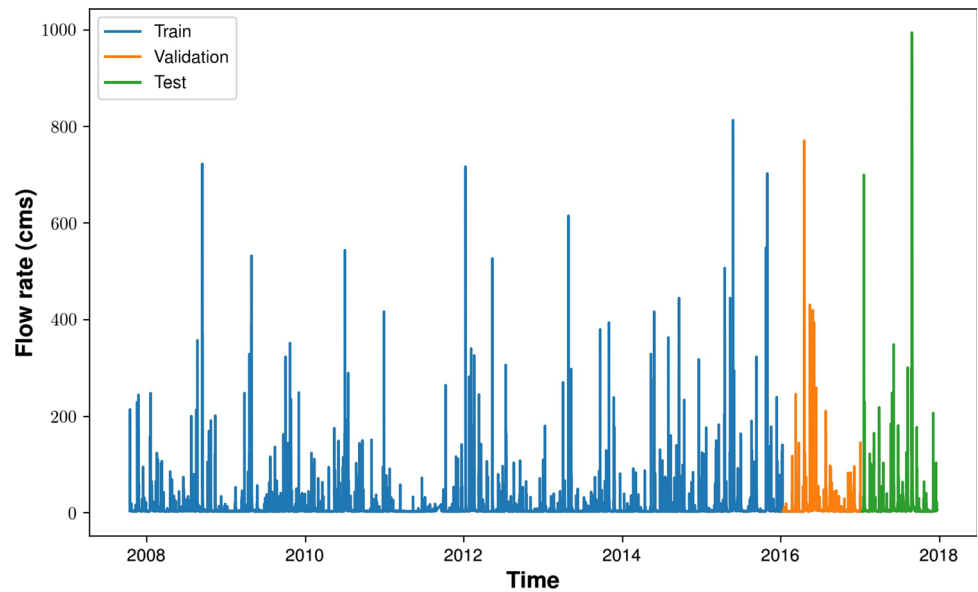
In machine learning, a mathematical model is built from existing data. However, the task of the machine learning model is to make predictions on future data that is not available at the model construction time. To evaluate the model performance on unseen data, a common practice in supervised machine learning is to split the data into three data sets which are used in different stages of the creation of the model.

Specifically, the model is initially fit on a training dataset, that is a set of examples used to fit the parameters (e.g., weights of connections between neurons of artificial neural networks) of the model. Successively, the fitted model is used to predict responses for the observations in a second dataset called the validation dataset (e.g., predict hydrograph given precipitation in this study). The validation dataset provides an unbiased evaluation of the model fit on the training dataset while tuning the model's hyperparameters (e.g., the number of the hidden units in neural network, number of LSTM layers, regularization, etc.). The combination of the hyperparameters with the best validation performance is then chosen for the machine learning model. Finally, the test dataset is used to provide an unbiased evaluation of a final model. If the examples from the test dataset have never been revealed to the model during training and validation stages, the test dataset is also called a holdout dataset.

Hydrologic data were split into train, validation, and test data sets. As shown in Fig. 3, all 15-min data up to the end of 2015 (2007–2015) were used for training. The entire year 2016 was used for validation and 2017 was used as the holdout test dataset. This train-validation-test split scheme is designed to minimize over-fitting and consistent with realistic prediction scenarios.

Particularly at the training stage, each time series is considered as a training example. A training *iteration* includes a forward propagation of the training example that computes the output, a backward propagation that computes the gradient, and an optimization step that updates learned parameters. Multiple training examples can be put

**Fig. 3** Training, validation, and test split



into a *batch* where both forward and backward propagation are processed in parallel respectively. One forward pass and one backward pass of all the training examples is called an *epoch*. More details can be found in [30].

During training, each sequence of data from the training dataset was input into the LSTM network as one batch. Built-in Adam optimization algorithm was used to optimize the MSE loss function. To avoid over-fitting, an  $l_2$  regularization with coefficient of  $10^{-6}$  was added for all learnable parameters. The learning rate was set to  $10^{-4}$ . These hyper-parameters were used for all tests for this study unless otherwise indicated.

To handle the missing data, a threshold of 90-min was set. If the missing data gap was less than or equal to 90 min (6 missing points), the missing points were imputed by linear interpolation. For gaps greater than 90 min, the sequence was split at the gaps. To speed up training and avoid the gradient exploding problem [44], the time series is further split into even shorter series.

To help the data-driven model extrapolate better, a minimum-maximum scaling was applied to both input and output variables [20, 22] so that all transformed variables are in the range of [0, 0.9]. Preliminary tests showed the model trained by the transformed variables in the range of [0, 0.9] performed better in comparison to the one trained by variables in the range of [0, 1].

### 3.4 LSTM scenarios

#### 3.4.1 Scenario 1: Physical consistency of LSTM model

Physically, it is obvious that the amount and pattern of rainfall collected by gages near/upstream of the flow gage

are more relevant to runoff discharge than those collected by gages that are far from/downstream of the river gage. To explore how the LSTM model results are spatially distributed and verify that the trained data-driven model is consistent with the physical intuition, two numerical tests were conducted using precipitation data from all 153 aforementioned rainfall gages:

1. Precipitation data from each rainfall gage was used to train a separate LSTM model. Then, the training loss of all of the 153 models were recorded and compared. The assumption was: if the LSTM model could actually learn the physical correlation between precipitation and river discharge, the model trained with more relevant input data should perform better.
2. All gages' data were used to train a single LSTM model. It is natural to assume that a physically consistent model should pay more attention to the more *important* gages.

The purpose of these two numerical tests is to explore the characteristics of the LSTM network for the defined application and reduce the number of input gages to the model based on the physical intuition of the problem.

For the first test, each model was trained for 200 epochs on the training data set combined with the validation data set and the best performing epoch with the minimum training error was recorded. Because the models trained in this test was not intended to be used for prediction, regularization and validation were not applied in this case. For the second test, due to the higher input dimension more epochs were required for the model to converge. Thus, the all-gage model was trained for 400 epochs on the training data set. The best performing epoch on the validation data set was chosen as the trained model. Regularization and

validation were applied here to (1) cope with the ill-conditioning problem when highly correlated precipitation data from different gages are presented; (2) prevent overfitting so that the model parameters including the first layer weights, come from a meaningful model. As noted before, the first layer of the LSTM network takes precipitation input. After the training, all the learnable input-hidden weights ( $W_{ii}$ ,  $W_{if}$ ,  $W_{ig}$ , and  $W_{io}$ ) of the first LSTM layer are grouped by gages and then flattened to a vector  $W$ , i.e.,

$$W = [W_{ii,1}, W_{if,1}, W_{ig,1}, W_{io,1}, \dots, W_{ii,n}, W_{if,n}, W_{ig,n}, W_{io,n}] \quad (11)$$

where  $n$  is the number of hidden units, i.e., 10. For each gage, three parameters of the learnable input-hidden weights were defined by the  $l_1$ ,  $l_2$ , and  $l_\infty$  norms of  $W$ :

$$\|W\|_1 = \sum_{i=0}^n (|W_{ii}| + |W_{if}| + |W_{ig}| + |W_{io}|) \quad (12)$$

$$\|W\|_2 = \left[ \sum_{i=0}^n (W_{ii}^2 + W_{if}^2 + W_{ig}^2 + W_{io}^2) \right]^{1/2} \quad (13)$$

$$\|W\|_\infty = \max\{|W_{ii,1}|, |W_{if,1}|, |W_{ig,1}|, |W_{io,1}|, \dots, |W_{ii,n}|, |W_{if,n}|, |W_{ig,n}|, |W_{io,n}|\} \quad (14)$$

Thus, for each gage, this test generates four parameters: the training error  $e$ , and the three norms of the weight vector  $W$ . To find any correlation, if any, among the three norms of the weight vector and training errors from the first numerical test, a correlation analysis was conducted using both Pearson correlation coefficient ( $r$ ) and Spearman's rank correlation coefficient ( $\rho$ ).

### 3.4.2 Scenario 2: LSTM model using 10 rainfall gages

To reduce the training time and the need for input data, it is necessary to reduce the number of gages used for the training. In addition, reducing the number of gages should not decrease, if not increase, the performance of the model. An exhaustive feature selection would require trying all combinations of gages which means training  $2^{153}$  models which is infeasible. Thus, the choice of rainfall gage was the 10 most relevant gages determined by scenario 1 using the gages with the minimum training errors.

A slightly different training process for this test was followed since this model was intended to be used for prediction. The LSTM model was trained on the training data set and regularization was added. The number of epochs were restricted to 200 and the best performing (in terms of evaluation score) epoch on the validation data set was chosen as the training result.

To show the causal improvement of this feature selection approach, more numerical tests were conducted.

Comparison was made among models trained with the 10 best gages (based on training error), 10 randomly sampled gages (sampled 5 times) from all 153 gages, 10 randomly sampled gages within the watershed, and 10 closest gages to the discharge gage. Thus, a total of eight models using 10 rainfall gages were built and tested.

### 3.5 Analysis and comparison of LSTM and GSSHA models

Since this study focuses on RR prediction for flood events, the evaluation was focused on flood events instead of normal flow regime dominated by the base flow and tidal mechanisms. The Nash–Sutcliffe model efficiency (NSE) and root-mean-square error (RMSE) were selected as the metrics for model evaluation. As introduced in Sect. 3.2, our developed model (as well as GSSHA model) is not a time series forecasting model. Hence, the metrics do not include those evaluations for time series models such as persistency criterion which compares the forecasting with the predictions from a naive persistence model.

To make a fair comparison of the LSTM model and the benchmark GSSHA model, an additional LSTM model was trained with data from 2014/12/13 to 2015/01/30, the same period used to calibrate GSSHA model. Moreover, the LSTM model takes the same set of input rainfall gages as the GSSHA model so that the two models have access to the exact same information. This LSTM model is referred as the reduced LSTM model for the rest of this paper. For comparison, four events of different scale in terms of precipitation and river discharge were chosen from the test data set (2017); low rainfall event from 9/28/2017 to 10/6/2017, moderate rainfall event from 12/16/2017 to 12/19/2017, high rainfall event from 12/2/2017 to 12/12/2017, and finally, an extreme rainfall event including Hurricane Harvey, which started from August 23 and ended on September 1, 2017. Note that the moderate event follows the high event with an interval of 4 days. Here, it was assumed the precipitation of the first event had completed runoff by the time the second event starts. Again, the choice of these four events were due to the limitation of computational expense of the GSSHA model, not the LSTM model. The LSTM model has no such restriction and was tested for all of 2017's flood events.

## 4 Result and discussion

The results and discussion section will start by presenting the results from the developed LSTM model, how well the model can predict runoff discharge using precipitation from each gage, and how consistent the results were with physical intuition. For this purpose, the two numerical tests



of experiment 1 will be discussed. Next, the performance of the constructed LSTM model with the 10 selected gages' data as input will be investigated and the results will be compared with the model using all 153 gages' data as input. Finally, the results of the GSSHA model developed for the Brays Bayou watershed will be presented and the difference between the predictions from GSSHA and LSTM will be discussed.

#### 4.1 Physical consistency of LSTM result

The training errors of LSTM models using each single rainfall gage (first numerical test in experiment 1) are shown in Fig. 4. The lowest training error was 29.94 in a gage just upstream of the discharge gage and highest training error was 278.11 in a gage located outside of Harris County. From Fig. 4 it can be seen that gages with the best performance are the ones located within or near the watershed. In fact, the Pearson correlation between the training error extracted from the LSTM model and the physical distance between the rainfall gages and the USGS gage was significant with a  $p$  value of  $3.5E-31$  and  $r = 0.77$ . This results show that similar to the process-

driven model, where precipitation drives runoff and the amount of precipitation falls into the watershed is represented by the interpolation of the rainfall gage recording, the data-driven model also performs better when better representation of the distributed precipitation is provided.

The correlation between training error  $e$  and the three parameters in scenario 1 could provide intuition on how much attention the LSTM model pays to the important rainfall gages. This makes more sense when considering the fact that the spatial distribution of the *best performing gages* matches very well with the physical intuition. The Pearson correlation coefficient, Spearman's rank correlation coefficient, and the respective  $p$  values of the statistical  $t$  tests are shown in Table 1. All statistical  $t$  tests suggest it is safe to reject the null hypothesis that the weight parameter is uncorrelated with the gage training error. Moreover, there is a non-trivial negative correlation between the norms of first layer weights and performance of model trained using the corresponding gage. The result suggests that statistically the LSTM model pays more attention to the *physically important* gages than those irrelevant gages.

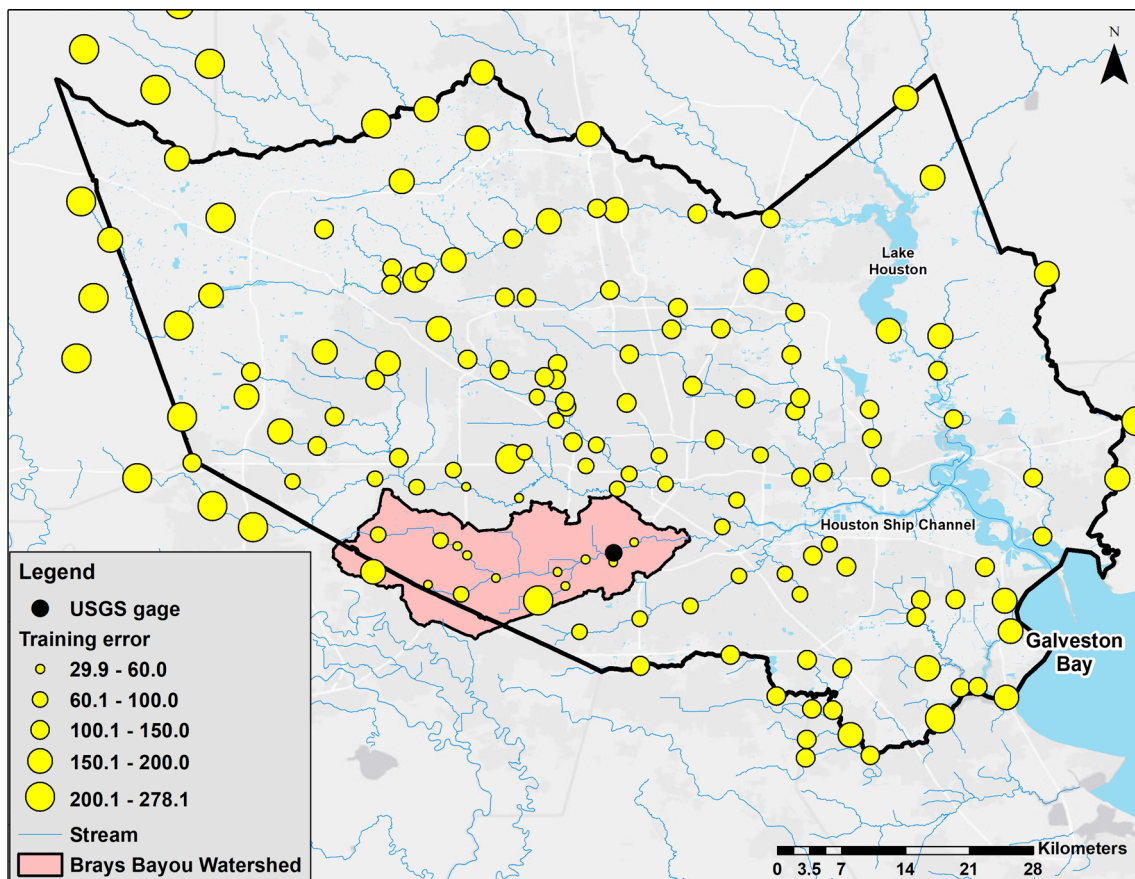


Fig. 4 Training error map for single gage models

**Table 1** Correlation between gage training error and weights parameters

Parameter	$r$	$p$ value of $r$	$\rho$	$p$ value of $\rho$
$\ W\ _1$	- 0.731	7.551e-27	- 0.335	2.264e-5
$\ W\ _2$	- 0.742	4.457e-28	- 0.343	1.389e-5
$\ W\ _\infty$	- 0.664	9.071e-21	- 0.318	6.097e-5

As suggested in [1, 31], eliminating redundant gages effectively improves the predictions of RR models. The consistency not only suggests the LSTM model is paying more attention to the more important gages, but also provides an efficient way of choosing rainfall gages for the LSTM model. Unlike the dedicated studies of choosing rainfall gages for RR modeling using areal rainfall optimization [1], the LSTM model can provide a coarse yet fast approach to pick the most relevant gages.

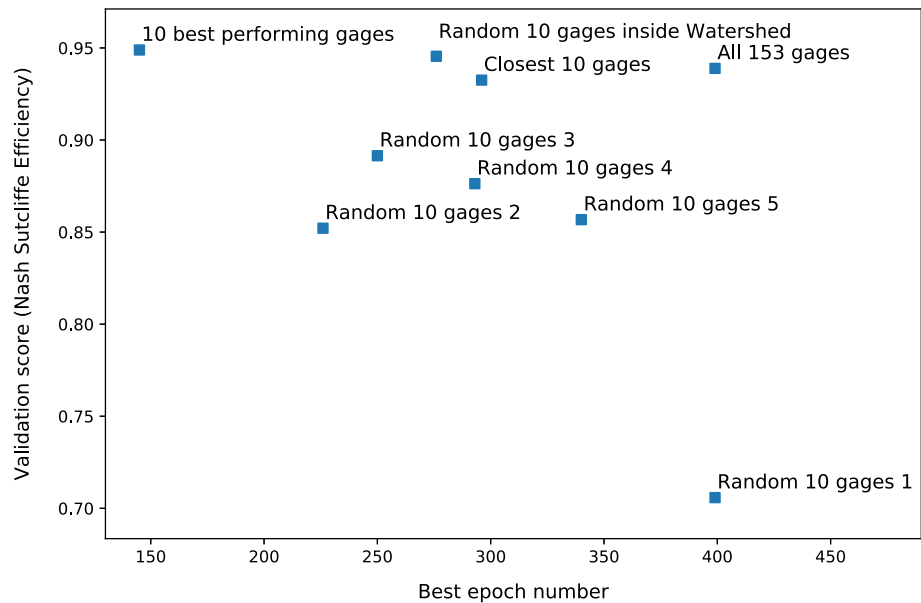
### 4.2 LSTM prediction

As noted before, an exhaustive feature selection among the rainfall gages is infeasible. Using the feature selection criterion suggested in scenario 1, the 10 gages with the

lowest training error in the first test, which are mostly located inside the watershed (the only gage outside the watershed is also very close to the watershed boundary), were picked as the input of the finalized LSTM model. To show the causal improvement of feature selection, we also compared the model performance (see Fig. 5) with models using (1) 10 randomly chosen gages within Harris County (sampled 5 times) (2) 10 randomly chosen gages within the watershed (3) closest 10 gages. The result in Fig. 5 shows that the best 10 gages model not only has the best validation score, but also converged faster than all other aforementioned models. Note that the model with 10 randomly chosen gages within the watershed also has a high validation score (0.945) but this model shares 6 common gages with the best 10 gages model. Nonetheless, its best epoch is 296 which is more than twice as much as that of the best 10 gages model indicating longer computational time for training.

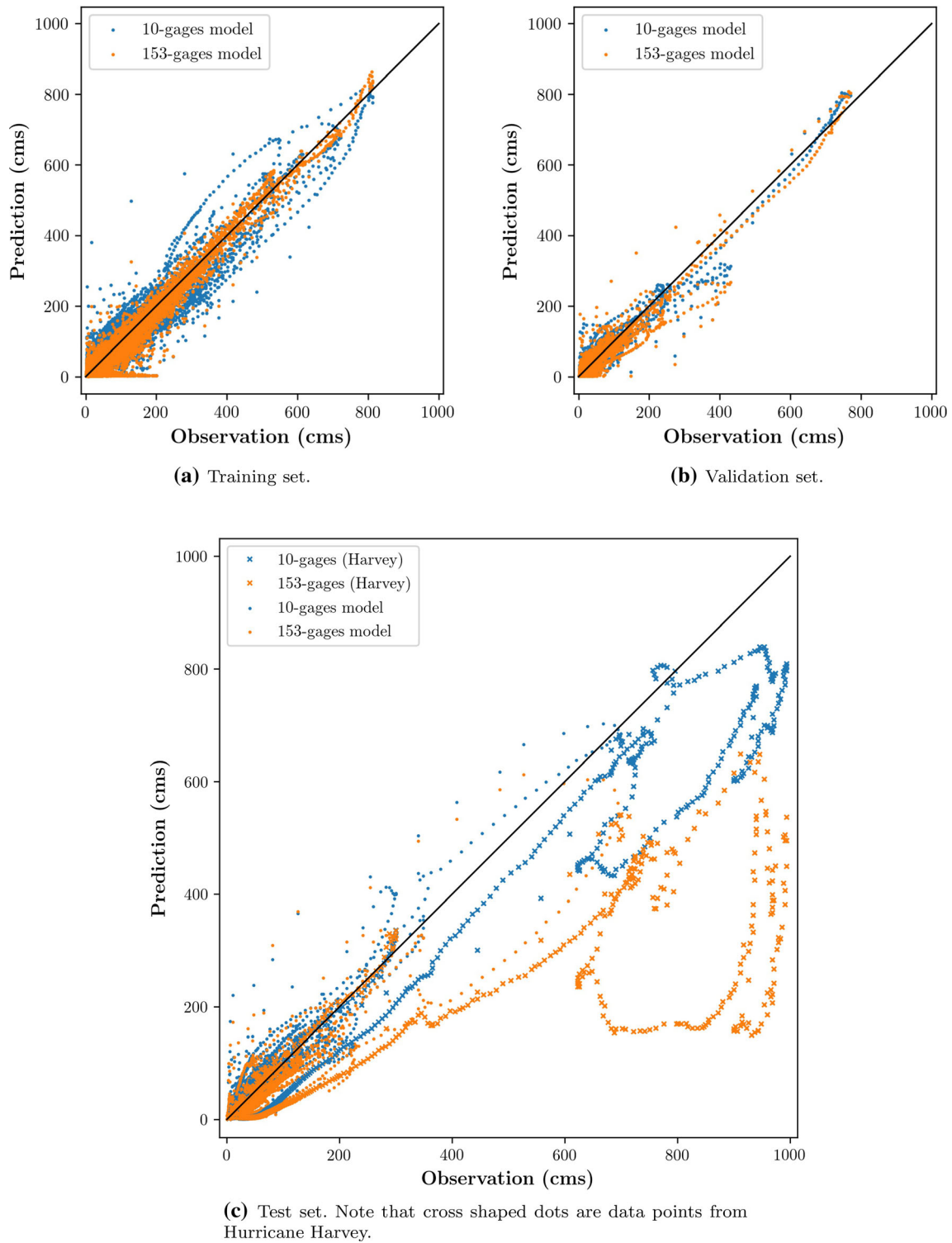
Compared to the 153-gage model in experiment 1, training of the best 10 gages model converged significantly faster: It took 145 epochs to converge, whereas the 153-gage model required more than 350 epochs to converge using the same learning rate. The convergence analysis of Adam optimization algorithm is out of the

**Fig. 5** Validation score versus number of epochs



**Table 2** Evaluation scores of 10-gages model versus 153-gages model

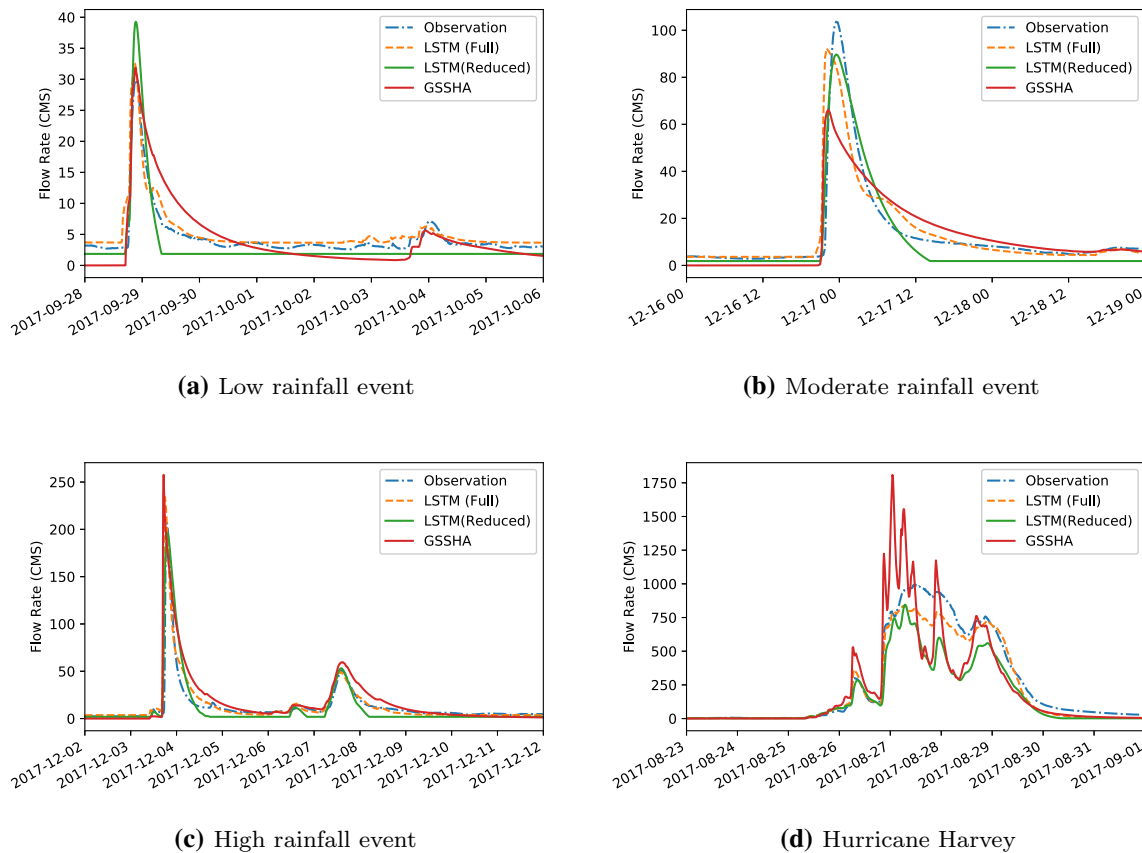
Data set	10-gages model		153-gages model	
	RMSE	NSE	RMSE	NSE
Training	8.11	0.921	5.65	0.961
Validation	7.83	0.947	8.49	0.938
Test	17.62	0.942	42.24	0.666
Test excluding Hurricane Harvey	8.32	0.906	10.04	0.864



**Fig. 6** Scatter plot of prediction versus observation for 10-gages and 153-gages models

scope of this paper; however, it is clear that the convergence of Adam algorithms is dimension dependent [27]. Besides, lower dimension implies lower computational cost for each iteration. Hence, training models with fewer inputs would be more efficient.

The evaluation scores (RMSE and NSE) were computed for both 10-gages model and 153-gages model on training/validation/test data set and are shown in Table 2. The 153-gages model had lower training error but higher



**Fig. 7** Comparison of the ground truth flow rates and predicted flow rates computed by GSSHA and LSTM. The distinct initial gap between prediction of GSSHA model and observation shown in **a, b**,

validation/test error compared to the 10 gages model which implies more over-fitting of the 153-gages model.

Moreover, the test error of both models were significantly higher than the training and validation errors. However, this behavior is explainable and does not indicate our model is over-fitting. The larger test error was dominated by under-predicting Hurricane Harvey which was included in the test set (see Fig. 6c). It should be noted that, Hurricane Harvey was an extraordinary flooding event in which, due to the high volume of precipitation, inter-basin transfer happened in many of the watersheds in the Greater Houston Area. Such phenomena is almost impossible to capture even with process-based models when only one watershed is modeled. Considering the uniqueness and rarity of Hurricane Harvey and its different behavior in both precipitation pattern and volume, the prediction of the 10-gages model on Hurricane Harvey, as shown in Fig. 7, is acceptable. The test result actually shows the relatively good extrapolation ability of the data-driven model.

From Fig. 6, it can be seen that the test set contains target discharge above 900 cm, while the training/validation sets have lower peak flow rate. The 10-gages model clearly performs better than the 153-gages model on an

is due to cold starting GSSHA simulation without an initial condition. Such a gap becomes invisible as event scale increases (**c, d**)

extreme event (Harvey). The flow rate versus time plots of both models for every event with peak flow larger than 30 cm are shown in Fig. 9 in “Appendix”. It can be shown that for the majority of the events except Harvey, both models are making reasonable predictions. Table 2 shows that if the time series containing Hurricane Harvey was excluded from the test set, the performance of the 10-gages model would be closer to those on the training and validation set. However, compared to the 10-gages model, the 153-gages model still seems to have larger variance given that it has better training score (compared to 10-gages model) but worse validation/test score. Due to the superior performance of the 10-gage LSTM model chosen through feature selection, this model was used as the final data-driven model in this study. Thus, for the rest of this paper, the 10-gage model is referred as the full LSTM model unless indicated otherwise.

### 4.3 Comparison of calibrated GSSHA and LSTM result

The evaluation metrics of the GSSHA model, reduced LSTM model, and full LSTM model are shown in Table 3.

**Table 3** Prediction performance comparison of GSSHA and LSTM model on selected events

Event	GSSHA		Full LSTM		Reduced LSTM	
	RMSE	NSE	RMSE	NSE	RMSE	NSE
Low event	2.51	0.636	1.43	0.881	2.34	0.684
Moderate event	10.49	0.725	8.99	0.798	5.49	0.925
High event	17.62	0.433	12.00	0.737	9.88	0.822
Harvey	185.64	0.700	69.73	0.958	169.42	0.751

Compared to GSSHA, the LSTM models show superior prediction performance in every test event. Note that in low and high events, GSSHA had lower than 0.7 NSE since it over-predicted the peak flow and predicted a delayed runoff. The reduced LSTM model is particularly good at predicting the moderate and high event with significantly better evaluation scores compared to both GSSHA model and full LSTM model. The superiority over the full LSTM model is probably due to the biased training set of the reduced LSTM model. The reduced training set consists of rainfall events of similar scale (medium to high). As the peak flow rate continues to increase, the full LSTM starts to stand out. From Fig. 7, we can clearly see that the full LSTM model prediction closely follows the observed flow rate.

One possible approach to improve the performance of the GSSHA model is to give it more degrees of freedom. For instance, instead of assuming all the river channels in the watershed have the same Manning's coefficient, we can allow river branches to have different Manning's coefficients and find the optimal coefficients by more extensive calibration. However, without a more efficient optimization procedure, calibration of GSSHA can be significantly more time consuming. The fastest forward simulation of GSSHA on the calibration dataset takes 2 h 54 min, when training LSTM model on the exact same dataset for 400 epochs takes 7 min. And the entire calibration would require tens or hundreds of forward GSSHA runs depending on the dimension of calibration coefficients. Another potential solution would be improving the quality of geometry input variables, activating the infiltration module, and increasing the model spatial resolution. All of these solutions will

dramatically increase the cost of data acquisition, model set-up, and computation.

## 5 Conclusions

In this study, we have shown the potential use of long-short-term-memory networks (LSTM) for RR modelling, using 15-min discharge and precipitation data for the first time. The experimental results from real world hydrologic data validated that the data-driven model can not only accurately predict stream flow given precipitation as the sole input when the scales of test data and training data are identical (interpolation), but also extrapolate well as shown in the prediction of Hurricane Harvey (NSE at 0.958 and RMSE at 69.73 cm for an event with peak flow at 993.92 cm). When compared to the process-driven model, GSSHA, the data-driven model is clearly more efficient and robust in terms of prediction and calibration. A forward GSSHA simulation of a 47 days event takes more than 2 h to run, but the prediction of the same event can be generated within a second by an LSTM model. Due to numerical stability restrictions, GSSHA simulation of 1 month period that contains an event like Hurricane Harvey could take more than 24 h. In addition, we explored the interpretability of the LSTM model in terms of its attention distribution on the input space. We found that the gage attention described in this paper can be used as a coarse yet fast measure to select gages in flood prediction. The experimental result shows that using the gage attention criterion is better than selecting the closest gages.

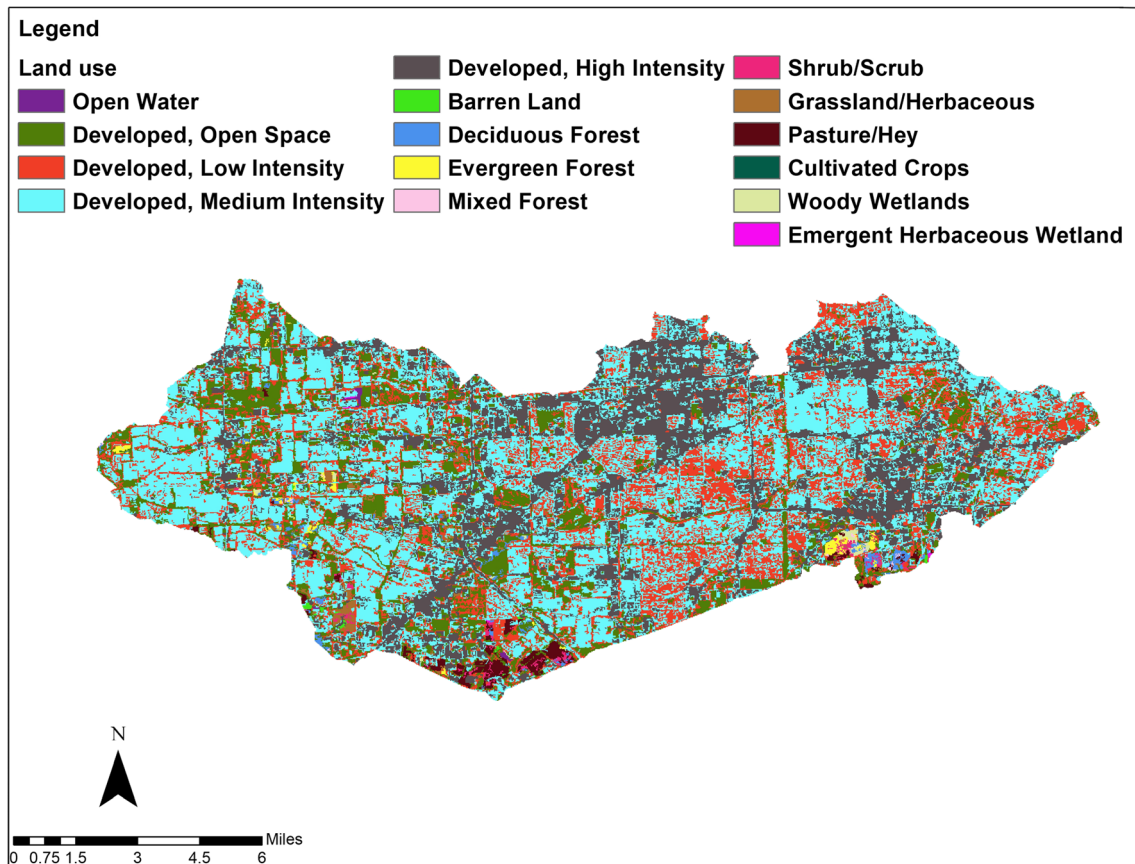
**Acknowledgements** This research was funded by the Severe Storm Prediction, Education and Evacuation from Disasters Center (Grant No. R09252) and the National Oceanic and Atmospheric Administration (Grant No. NA18NOS0120158). Their support is gratefully acknowledged.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## Appendix

See Figs. 8, 9 and Table 4.



**Fig. 8** Land use of study area

**Fig. 9** Comparison of observation, 10-gages LSTM, and 153-gages LSTM. A close look at the lower discharge region would reveal the unphysical oscillation of 153-gages prediction, indicating overfitting

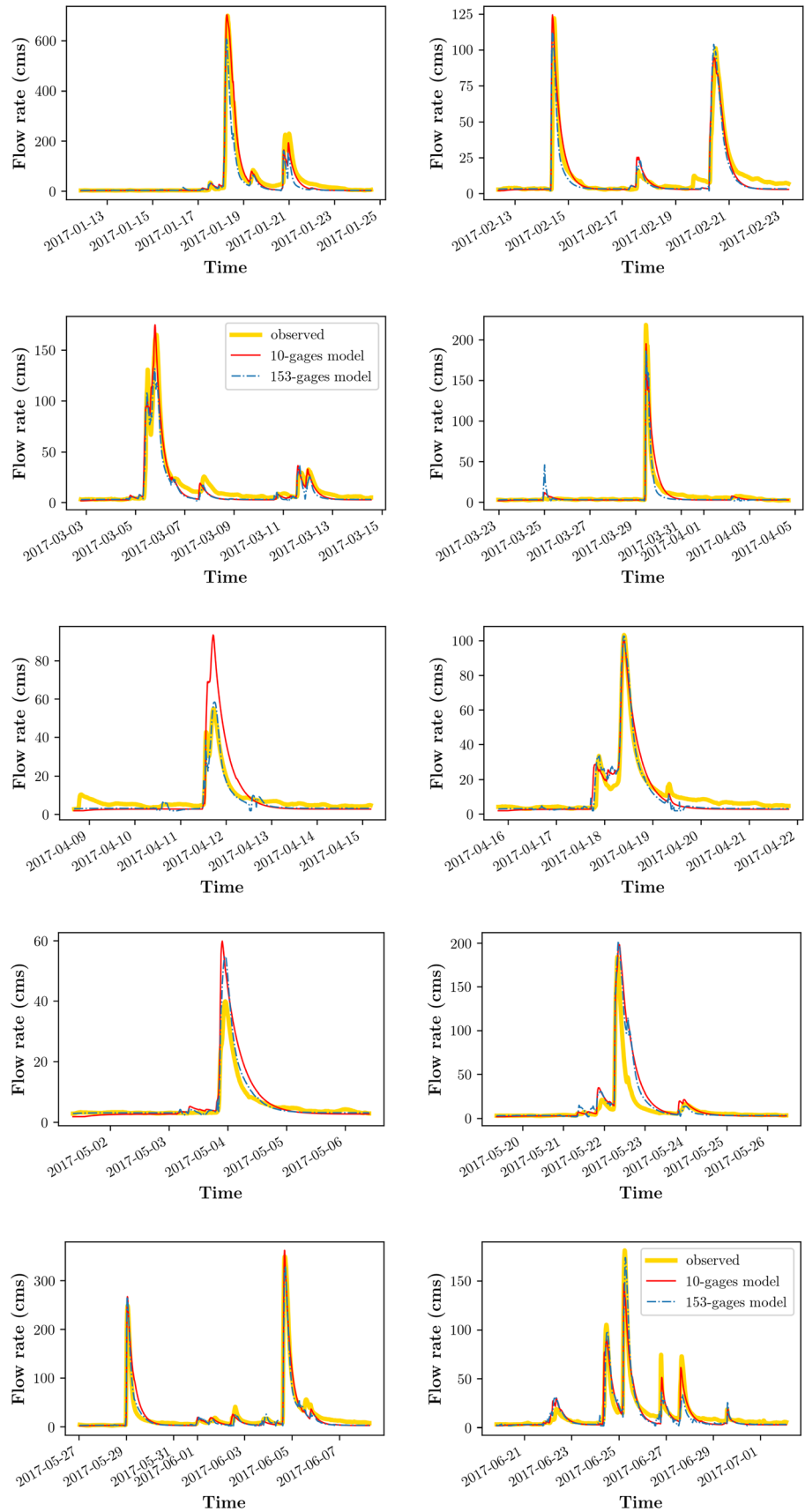
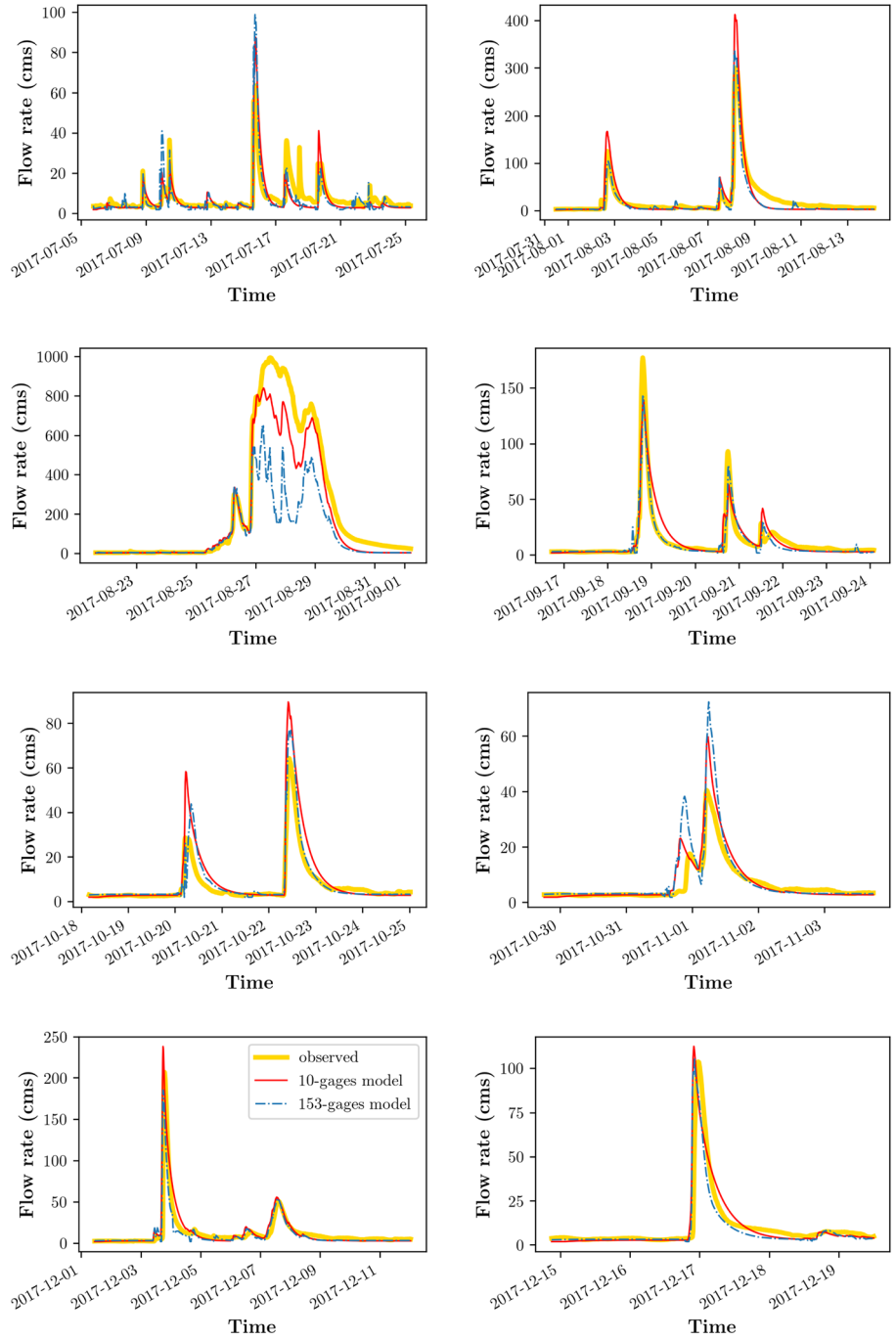


Fig. 9 continued





**Table 4** Mannings  $n$  values for various land covers to use for GSSHA simulation

NLCD value	Normal manning's $n$ value	Allowable range of $n$ values
11	0.040	0.025–0.05
21	0.040	0.03–0.05
22	0.100	0.08–0.12
23	0.080	0.06–0.14
24	0.150	0.12–0.20
31	0.025	0.023–0.030
41	0.160	0.10–0.16
42	0.160	0.10–0.16
43	0.160	0.10–0.16
52	0.100	0.07–0.16
71	0.035	0.025–0.050
81	0.030	0.025–0.050
82	0.035	0.025–0.050
90	0.120	0.045–0.15
95	0.070	0.05–0.085

## References

- Anctil F, Lauzon N, Andréassian V, Oudin L, Perrin C (2006) Improvement of rainfall–runoff forecasts through mean areal rainfall optimization. *J Hydrol* 328(3–4):717–725
- Bedient PB, Hoblit BC, Gladwell DC, Vieux BE (2000) Nexrad radar for flood prediction in houston. *J Hydrol Eng* 5(3):269–277
- Bengio Y, Simard P, Frasconi P et al (1994) Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw* 5(2):157–166
- Borah DK, Ahmadisharaf E, Padmanabhan G, Imen S, Mohamoud YM (2018) Watershed models for development and implementation of total maximum daily loads. PhD thesis, American Society of Civil Engineers
- Brays Bayou (2019). <https://www.hcfcd.org/projects-studies/brays-bayou/>. Accessed 7 Mar 2019
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32. <https://doi.org/10.1023/A:1010933404324>
- Brunner GW (1997) HEC-RAS (river analysis system). In: North American water and environment congress and destructive water. ASCE, New York, pp 3782–3787
- Daniel EB, Camp JV, LeBoeuf EJ, Penrod JR, Abkowitz MD, Dobbins JP (2011) Watershed modeling using GIS technology: a critical review. *J Spat Hydrol* 10(2):13–28
- Dawson CW, Wilby R (1998) An artificial neural network approach to rainfall–runoff modelling. *Hydrol Sci J* 43(1):47–66. <https://doi.org/10.1080/02626669809492102>
- Doshi-Velez F, Kim B (2017) Towards a rigorous science of interpretable machine learning. Preprint [arXiv:1702.08608](https://arxiv.org/abs/1702.08608)
- Downer CW, Ogden FL (2004) GSSHA: model to simulate diverse stream flow producing processes. *J Hydrol Eng* 9(3):161–174. [https://doi.org/10.1061/\(ASCE\)1084-0699\(2004\)9:3\(161\)](https://doi.org/10.1061/(ASCE)1084-0699(2004)9:3(161))
- Downer CW, Ogden FL (2006) Gridded surface subsurface hydrologic analysis (GSSHA) user's manual; version 1.43 for watershed modeling system 6.1. Technical report, Engineer Research and Development Center, Vicksburg, MS, Coastal and Hydraulics Laboratory
- Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall–runoff models. *Water Resour Res* 28(4):1015–1031
- Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. *IEEE Trans Pattern Anal Mach Intell* 35(8):1915–1929
- Faur C, Cougnaud A, Dreyfus G, Le Cloirec P (2008) Modelling the breakthrough of activated carbon filters by pesticides in surface waters with static and recurrent neural networks. *Chem Eng J* 145(1):7–15
- Fry T, Maxwell R (2018) Using a distributed hydrologic model to improve the green infrastructure parameterization used in a lumped model. *Water* 10(12):1756
- Furl C, Ghebreyesus D, Sharif H (2018) Assessment of the performance of satellite-based precipitation products for flood events across diverse spatial scales using gssha modeling system. *Geosciences* 8(6):191
- Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: continual prediction with LSTM. In: 9th international conference on artificial neural networks, pp 850–855
- Harris County Flood Warning System (2019). <https://www.harriscountyfws.org/>. Accessed 7 Mar 2019
- Hettiarachchi P, Hall M, Minns A (2005) The extrapolation of artificial neural networks for the modelling of rainfall–runoff relationships. *J Hydroinform* 7(4):291–296
- Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncert Fuzziness Knowl Based Syst* 6(02):107–116
- Hsu KI, Gupta HV, Sorooshian S (1995) Artificial neural network modeling of the rainfall–runoff process. *Water Resour Res* 31(10):2517–2530
- Inc ST (2002) Channel interconnected and model pond routing version 3.10 (2002). <http://www.streamnologies.com/products/icpr/icpr.htm>. Accessed 7 Mar 2019
- Johannet A, Borrell Estupina V, Pistre S et al (2015) Neural networks for Karst groundwater management: case of the Lez spring (Southern France). *Environ Earth Sci* 74(12):7617–7632
- Kan G, Yao C, Li Q, Li Z, Yu Z, Liu Z, Ding L, He X, Liang K (2015) Improving event-based rainfall–runoff simulation using an ensemble artificial neural network based hybrid data-driven model. *Stoch Env Res Risk Assess* 29(5):1345–1370. <https://doi.org/10.1007/s00477-015-1040-6>
- Kao IF, Zhou Y, Chang LC, Chang FJ (2020) Exploring a long short-term memory based encoder–decoder framework for multi-step-ahead flood forecasting. *J Hydrol* 2020:124631
- Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. Preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- 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 Discuss*. <https://doi.org/10.17605/OSF.IO/QV5JZ>
- Le XH, Ho HV, Lee G, Jung S (2019) Application of long short-term memory (LSTM) neural network for flood forecasting. *Water* 11(7):1387
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
- Lindström G, Johansson B, Persson M, Gardelin M, Bergström S (1997) Development and test of the distributed HBV-96 hydrological model. *J Hydrol* 201(1–4):272–288

32. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. Preprint [arXiv:1506.00019](https://arxiv.org/abs/1506.00019)
33. Lü H, Hou T, Horton R, Zhu Y, Chen X, Jia Y, Wang W, Fu X (2013) The streamflow estimation using the Xinanjiang rainfall–runoff model and dual state-parameter estimation method. *J Hydrol* 480:102–114. <https://doi.org/10.1016/j.jhydrol.2012.12.011>
34. Mhammedi Z, Hellicar A, Rahman A, Kasfi K, Smethurst P (2016) Recurrent neural networks for one day ahead prediction of stream flow. In: Proceedings of the workshop on time series analytics and applications—TSAA'16, pp. 25–31. <https://doi.org/10.1145/3014340.3014345>. <http://dl.acm.org/citation.cfm?doid=3014340.3014345>
35. Nayak P, Sudheer K, Ramasastri K (2005) Fuzzy computing based rainfall–runoff model for real time flood forecasting. *Hydrol Process Int J* 19(4):955–968
36. Nerrand O, Roussel-Ragot P, Personnaz L, Dreyfus G, Marcos S (1993) Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms. *Neural Comput* 5(2):165–199
37. NOAA National Centers for Environmental Information (NCEI) (2018) US billion-dollar weather and climate disasters. <https://www.ncdc.noaa.gov/billions/>. Accessed 23 May 2018
38. Pappenberger F, Beven KJ, Hunter N, Bates P, Gouweleeuw B, Thielen J, De Roo A (2005) Cascading model uncertainty from medium range weather forecasts (10 days) through a rainfall–runoff model to flood inundation predictions within the European flood forecasting system (EFFS). *Hydrol Earth Syst Sci Dis* 9(4):381–393
39. Paszke A, Chanan G, Lin Z, Gross S, Yang E, Antiga L, Devito Z (2017) Automatic differentiation in PyTorch. In: 31st conference on neural information processing systems (Nips), pp 1–4. <https://doi.org/10.1017/CBO9781107707221.009>
40. Silva-Araya W, Santiago-Collazo F, Gonzalez-Lopez J, Maldonado-Maldonado J (2018) Dynamic modeling of surface runoff and storm surge during hurricane and tropical storm events. *Hydrology* 5(1):13
41. Somu N, MR GR, Ramamritham K (2020) A hybrid model for building energy consumption forecasting using long short term memory networks. *Appl Energy* 261:114131
42. Srinivasulu S, Jain A (2006) A comparative analysis of training methods for artificial neural network rainfall–runoff models. *Appl Soft Comput* 6(3):295–306
43. Sudheer KP, Gosain AK, Ramasastri KS (2002) A data-driven algorithm for constructing artificial neural network rainfall–runoff models. *Hydrol Process* 16(6):1325–1330. <https://doi.org/10.1002/hyp.554>
44. Sutskever I, Vinyals O, Le Q V (2014) Sequence to Sequence Learning with Neural Networks. In: Ghahramani Z, Welling M, Cortes C, et al (eds) *Advances in Neural Information Processing Systems* 27. Curran Associates, Inc., pp 3104–3112
45. Talchabhadel R, Shakya NM, Dahal V, Eslamian S (2015) Rainfall–runoff modelling for flood forecasting (a case study on west rapti watershed). *J Flood Eng* 6(1):53–61
46. Taver V, Johannet A, Borrell-Estupina V, Pistre S (2015) Feed-forward vs. recurrent neural network models for non-stationarity modelling using data assimilation and adaptivity. *Hydrol Sci J* 60(7–8):1242–1265
47. Torres JM, Bass B, Irza N, Fang Z, Proft J, Dawson C, Kiani M, Bedient P (2015) Characterizing the hydraulic interactions of hurricane storm surge and rainfall–runoff for the Houston–Galveston region. *Coast Eng* 106:7–19. <https://doi.org/10.1016/j.coastaleng.2015.09.004>
48. van Gerven M, Bohte S (2018) Artificial neural networks as models of neural information processing. *Frontiers Media*, New York
49. Wu Y, Liu Z, Xu W, Feng J, Palaiahnakote S, Lu T (2018) Context-aware attention LSTM network for flood prediction. In: 2018 24th international conference on pattern recognition (ICPR), pp 1301–1306
50. Young CC, Liu WC (2015) Prediction and modelling of rainfall–runoff during typhoon events using a physically-based and artificial neural network hybrid model. *Hydrol Sci J* 60(12):2102–2116
51. 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(8):2199–2212
52. Zarzar CM, Hosseiny H, Siddique R, Gomez M, Smith V, Mejia A, Dyer J (2018) A hydraulic multimodel ensemble framework for visualizing flood inundation uncertainty. *JAWRA J Am Water Resour Assoc* 54(4):807–819
53. Zhijia L, Lili W, Hongjun B, Yu S, Zhongbo Y (2008) Rainfall–runoff simulation and flood forecasting for Huaihe Basin. *Water Sci Eng* 1(3):24–35

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