



Exploring the best sequence LSTM modeling architecture for flood prediction

Wei Li¹ · Amin Kiaghadi¹ · Clint Dawson¹

Received: 28 May 2020 / Accepted: 2 September 2020 / Published online: 19 September 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. Recently, the success of the recurrent neural network (RNN) applied to sequential models has motivated groups to pursue RR modeling using RNN. Existing RNN based methods generally use either sequence input single output or unsynced sequence input and output architectures. In this paper, we propose a synced sequence input and output long short-term memory (LSTM) network architecture for hydrologic analysis and compare it to existing methods (sequence input single output LSTM). We expect the model will improve RR prediction in terms of accuracy, calibration training time, and computational cost. The key idea is to efficiently learn the long term dependency of runoff on past rainfall history. To be more specific, we use the indigenous ability of the LSTM network to preserve long term memory instead of artificially setting a time window for input data. In this way, we can avoid losing long term memory of the input, the calibration of the time window length, and excessive computation. The whole procedure mimics the traditional process-driven methods and is closer to the physics interpretation of the RR process. We conducted experiments on real-world hydrologic data from the Brays Bayou in Houston, Texas. Extensive experimental results clearly validate the effectiveness of our proposed method in terms of various statistical and hydrological related evaluation metrics. Notably, our experiment shows that some rainfall events could affect the runoff process in the test watershed for at least a week. For fine temporal resolution prediction, this long term effect needs to be carefully handled, and our proposed method is superior in this case.

Keywords Recurrent neural network · Hydrologic analysis · Sequence modeling · Long short-term memory network

1 Introduction

In the United States, flooding is the number one cause of natural-disaster losses with estimated annual damage of eight billion dollars [1]. Thus, reliable predictive tools for rainfall–runoff (RR) modeling are crucial for flood prevention, mitigation, and management. The literature is replete with studies using different modeling approaches to predict runoff (discharge) caused by a rainfall event [2–5]. Both process-driven [6–8] and data-driven [9, 10] approaches were applied in this research area. The results of the process-driven models are more realistic and scalable due to the use of analytical and empirical formulae based on

physical phenomena. However, extensive meteorological and geometric data requirements, skilled users, and continuous calibration processes make this class of RR models more obsolete. In contrast, data-driven models that rely on interpolating and extrapolating of data have become more popular for RR modeling. Nevertheless, the lack of knowledge of statistics and ML and using data-driven models as a black box could lead to dramatic errors.

The application of machine learning (ML) has been dramatically increasing mainly due to the significant increase in computing power and data availability. In hydrology, in particular, ML-based data-driven models have been widely used for streamflow prediction with feed-forward artificial neural network (ANN) being the most popular algorithm [9–13]. More recently, to address the limitations of feed-forward ANN in handling time-series data, recurrent neural network (RNN) algorithms have been used for RR modeling [14]. The loops inside the RNN

✉ Amin Kiaghadi
kiaghadi@utexas.edu

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

make it capable of capturing long-term dependencies in data. Theoretically, the hidden state inside RNN should be able to preserve the memory of past input data. However, in reality, standard RNN with artificial neurons as hidden units faces vanishing and exploding gradient issues for network training [15]. To this end, long short-term memory (LSTM) networks have been developed by adding cell state and gating mechanisms to the vanilla RNN [15, 16]. The gates within the LSTM network handle the decision process on forgetting or remembering the information by keeping the errors in memory, which avoids error signal decay [17, 18]. In other words, the gates within the LSTM help to preserve states and short-term dependencies over long periods. Although the LSTM network was introduced in 1997 by Hochreiter and Schmidhuber [17], it had not been used for RR modeling till 2016 [19]. Since 2016, a handful of studies have used LSTM for RR modeling and reported satisfying results [19–27]. Many of the studies mentioned above showed the superior performance of the LSTM network in capturing the dynamics of time-series compared to other RNN networks for hydrologic applications. Table 1 provides a summary of those studies and their architecture. As shown in Table 1, most of the studies used a fixed window size.

Part of the superiority of RNNs compared to traditional ANNs is due to their sequence regime of operation compared to fixed-size networks. Depending on the application of the network, different architectures, including sequence input and a single output (SISO), single input and sequence output, sequence input and sequence output, and synced sequence input and output (SSIO) can be used. While each of these structures are designed for a specific purpose, most of the studies that used the LSTM network for RR modeling have chosen sequence input and single output or sequence input and sequence output architectures

(Table 1). These architectures require determining a fixed window size, unlike the SSIO architecture that relies on the LSTM structure itself to capture the long dependencies.

Choosing a fixed window size forces LSTM to limit the long-term dependencies into the size of the selected window. On the other hand, the passage of hidden states from previous time steps in the SSIO architecture makes it capable of capturing the long-term dependencies on its own. In other words, there is no need to use LSTM (or even RNN) for RR modeling application if the user desire to use a fixed window regime since the LSTM network does not need to feed the model with the fixed window size. Additionally, using the fixed window size approach requires more memory because of the formation of a matrix with a size of batch size \times fixed window size \times number of input variables, compared to batch size \times number of input variables in the SSIO architecture, even though the two input dataset contains the same amount of information. Moreover, extensive knowledge of the watershed response to the rainfall events is required to determine the window size. While the choice of architecture is not very influential when working with a small dataset or coarse time resolution, it could make a significant difference in prediction accuracy, computational time, and storage requirements when increasing the time resolution or size of the study. The main goal of this study is to compare these two architectures with regards to prediction performance, computational time, and memory requirements.

The majority of the aforementioned studies could be categorized as time series-forecasting, meaning the flow at previous time steps (Q_{t-k}) is within the input vector to the model along with rainfall and/or other variables. Depending on the immediate past observations to make a prediction could be problematic in the case of damage to the observational flow gauge. Severe damages have been

Table 1 A summary of studies that used the LSTM network for rainfall–runoff modeling

Ref	Type	Time resolution	Type	Window size	Evaluation metric ^a
[23]	Time series	Daily	Sequence input and output	10	NSE, RMSE
[24]	Prediction	Daily	Sequence input and output	270	NSE, FHV, FMS, FLV
[25]	Time series	Hourly	Sequence input and output	1	R^2 , RMSE, NSE, MAE, ET_p , EQ_p
[21]	Time series	Hourly	Sequence input and output	32	RMSE
[19]	Time series	Daily	Sequence input and single output	200	NSE and RMSE
[20]	Prediction	Daily	Sequence input and single output	365	NSE, RMSE, r , α , β , FHV, FMS, FLV
[22]	Time series	Monthly	Sequence input and single output	5	MAE, R^2 , NSE, RMSE, bias, SD_{bias} , DISP
[26]	Time series	Hourly	Sequence input and output	8	RMSE, NSE, R^2
[27]	Prediction	NR ^b	Sequence input and single output	NR	MAPE, RMSE

^aPlease see Sect. 3.3 for the definitions

^bNR Not reported

reported to the gauges maintained by the National Oceanic and Atmospheric Administration (NOAA) and the United States Geological Survey (USGS) during hurricanes and severe storms [28]. Some of the damaged gauges have never come back to functionality. Furthermore, future development requires predicting runoff of hypothetical rainfall events with different return periods (scenario-based) as a part of flood management. Such a study would not be possible using the time series-forecasting approach due to the absence of observed discharge data. The few studies that did not use the Q_{t-k} as one of the inputs used extensive meteorological and watershed characteristics variables [20, 24]. The literature lacks an LSTM model that predicts runoff solely based on precipitation input.

In this study, we build LSTM models with different architectures that use high-temporal resolution rainfall (i.e., 15 min) as the sole input and generate runoff as the output. We will compare the SSIO with the SISO architecture with different window sizes. An extensive comparison among the architectures will be made through the use of various evaluation metrics that measure the performance of the models in predicting the hydrograph (runoff over time), computational time, and storage requirements.

2 Methodology

In this study, we aim to compare the two different recurrent neural network (RNN) architectures for RR modeling, which aims at predicting the streamflow hydrograph from precipitation input. We will first introduce the problem formulation. Then we will give a quick review of a basic neural network unit. In the end, we are going to show the two different architectures studied in this section.

2.1 Problem formulation

To formulate the RR prediction task in a data-driven setting, we use Q_t to denote the river flow rate at the outlet of a watershed at time step t , which is the quantity of interest (QoI) of this problem. Similarly, we use $X_t = \{X_{it}, i = 1, 2, \dots\}$ to denote the rainfall recordings at time step t in the watershed. For each gauge i , the precipitation recorded during the time step t is X_{it} . In practice, given a precipitation forecast, we would like to predict the river flow rate based on the precipitation forecast. Moreover, in flood management, we are also interested in predicting the runoff for hypothetical rainfall events where real-time observation does not exist. Thus, we wish to find the mapping $M : \{X_t, X_{t-1}, X_{t-2}, \dots\} \mapsto Q_t$ that could predict the river flow rate using past precipitation as the only input. In other words, we wish to predict future runoff (hydrograph) based on the history of rainfall (hyetograph).

In reality, RR is also affected by other physical processes such as evapotranspiration, infiltration, and snow melting that depend on other metrological data in addition to precipitation. In this study, however, we are focused on flood prediction with a high temporal resolution (every 15 min). Empirically rainfall drives the majority part of the runoff in this scenario. To sum up, the task of runoff modeling is to find the regression relationship between output river flow rate and input precipitation history.

2.2 Preliminaries

Long short-term memory (LSTM) In both our proposed model and the architecture we are comparing to, LSTM units are used. RNNs keep a hidden state vector, which changes according to the input at each time step. Theoretically, the hidden state vector preserves the memory of the history of the input, making RNNs a natural fit for our task. LSTM was proposed by [9] to deal with the exploding and vanishing gradient problems. The LSTM unit we use consists of a cell state c_t , an input gate i_t , a forget gate f_t , a cell gate g_t , and an output gate o_t . For each time step t with the precipitation input vector X_t , previous hidden cell state h_{t-1} , and previous cell state c_{t-1} , the updated hidden state h_t is computed by the following calculations:

$$i_t = \sigma(W_{ii}X_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (1)$$

$$f_t = \sigma(W_{if}X_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (2)$$

$$g_t = \tanh(W_{ig}X_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (3)$$

$$o_t = \sigma(W_{io}X_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * g_t \quad (5)$$

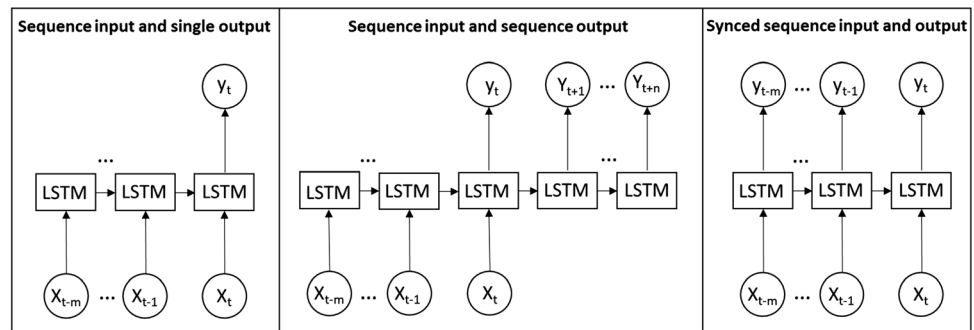
$$h_t = o_t * \tanh(c_t) \quad (6)$$

$\sigma(\bullet)$ is known as a sigmoid function. $*$ denotes Hadamard products. All W 's are weight matrices, and all b 's are bias matrices. Note that since we are working with real-world time series data, past time steps should not be affected by future time steps. Consequently, the bidirectional mechanism is not employed in this study. Both architectures use a single directional LSTM network.

2.3 LSTM architectures

The majority of existing LSTM RR literature have used sequence input single output (SISO) or sequence input/output model (see Fig. 1 for different structures). However, we argue that a synced sequence input/output (SSIO) model (input and output have the same length l) fits our task better. Note that l does not need to be fixed. Although the un-synced sequence input and output model has been used in hydraulic engineering, it is not perfectly suitable for our application in this study. The common use case

Fig. 1 Different LSTM architectures. It should be noted that $X_t = \{X_{it}, i = 1, 2, \dots\}$ where the precipitation recorded during the time step t is X_{it} to denote the rainfall recordings at time step t in the watershed



of the method is to predict runoff in multiple future time steps. However, our method aims at predicting the runoff at the next time step. Thus, we mentioned the architecture for introduction but did not discuss further in this paper.

For the sequence input single output and un-synced sequence input/output model, the length of the input vector is fixed from the training time. This setup has three potential disadvantages. (1) The excessive need to determine the best window size l_w . In other words, the window size itself becomes an extra hyperparameter and requires tuning. (2) The potential of losing long term rainfall history; if l_w is not large enough, the long term effect of rainfall cannot be captured within the time window. For flood prediction at a temporal resolution as high as every 15 min, there are 96-time steps for a single day.

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 beginning of the event. In this case, if l_w is set to be 100, the network will have trouble predicting a one-week event. (3) Compared to the synced sequence input/output model, the sequence input single output model requires more computation and memory to process the same time series. For instance, if we choose the window size to be 100, then the input data will be repeated 100 times.

Since we are focused on predicting the runoff with only precipitation input, the un-synced sequence input/output model is not applicable. To this end, we compare two different architectures: a synced sequence input/output model used by [29] and sequence input single output model used in [19–27].

3 Experiments and evaluation

In this section, we will introduce the experimental results based on a RR dataset from an urban watershed, Brays Bayou, in Houston, Texas.

3.1 Study area and dataset

Figure 2 shows the Brays Bayou watershed, Brays Bayou, and its tributaries located in the southwest of Harris County

and northeast of Fort Bend County, Texas, that we 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 and eventually to the Gulf of Mexico. 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 [1].

15-min precipitation data from 2007 to 2017 were collected from 15 rainfall gauges within the Brays Bayou watershed maintained by the Harris County Flood Control District (HCFCD, rainfall data is available at <https://www.harriscountyfws.org/>) and 15-min flow data were obtained from the United States Geological Survey (USGS) gauges. Within the Brays Bayou watershed, there are five flow gauges. In this study, only one freshwater gauge located very close to the watershed outlet (green circle in Fig. 2) was used to collect flow data for the purpose of training, validation, and test (discharge data is available at https://waterdata.usgs.gov/nwis/uv?site_no=08075000). Considering the RR process, predicting the flow at the outlet of the watershed is desirable not only for flood management purposes but also for any potential feeding to a storm surge model (flow flux). Thus, in this study, only the gauge located near the outlet of the watershed that is not influenced by the tide was selected. Hydrologic data (rainfall and river discharge) was split into the train, validation, and test data sets. All 15-min data up to the end of 2015 (2007–2015) was used for training. The entire year 2016 was used for validation, and 2017 was used as the holdout test dataset. This training-validation-test split scheme (shown in Fig. 3) is designed to minimize overfitting and be consistent with realistic prediction scenarios. We believe that the characteristics of the watershed that could affect runoff evolve in time. At prediction time, all the data we have will be in the past, and the prediction we aim to do is in the future. Thus, later data is used as test/validation than training.

Fig. 2 Brays Bayou watershed, streams, rainfall, and flow gauges. The figure on the bottom right shows rainfall and runoff during Hurricane Harvey (2017) (color figure online)

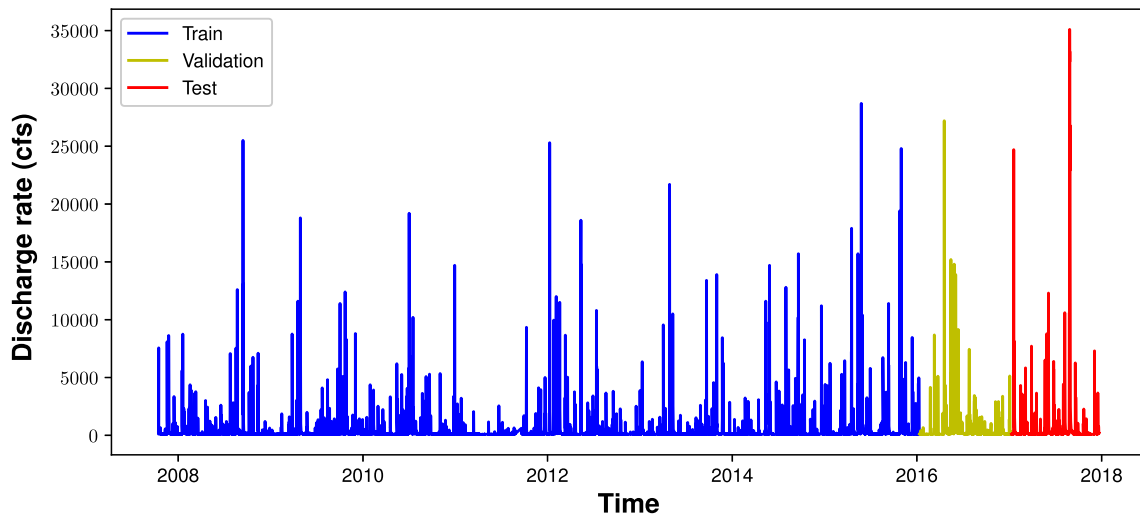
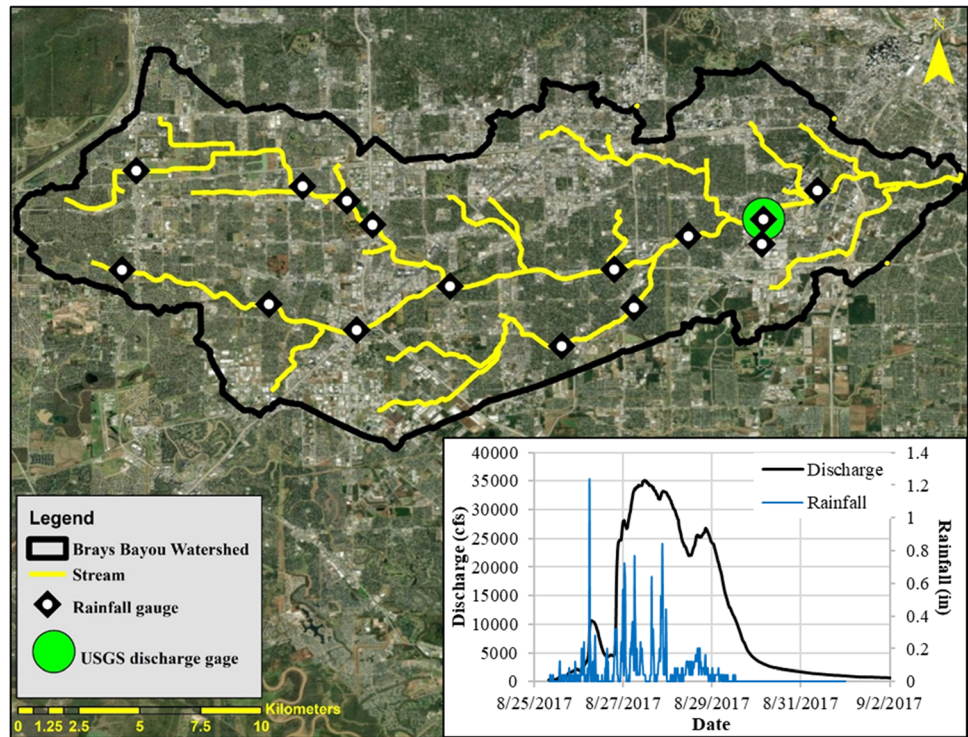


Fig. 3 Discharge rates used for training, validation, and test. Hurricane Harvey (2017) in the test dataset was an unseen event

3.2 Experimental setups

Here we introduce the detailed settings of our experiments, including the parameters for the compared models and details of the training stage.

3.2.1 Synced sequence input and output (SSIO) model

For the synced sequence input/output model, the training dataset originally consists of 16 columns. Each column is a time series of precipitation or river discharge spanning

from 2007 to the end of 2015. The training dataset contains several extreme events, as shown in Fig. 3, including Memorial Day Flood (2015) and Hurricane Ike (2008). The peak flow of the validation set is lower than that of the training set. The peak of the test dataset (from hurricane Harvey) is higher than the previous record and requires the model to extrapolate, but it is not high enough to be an outlier. Training an RNN on such a long sequence is challenging and problematic. Thus, the training dataset is split into shorter snippets, where each snippet starts from the beginning of a runoff event and ends at a definitive

conclusion of a runoff event. The snippets do not necessarily have the same length. During training each snippet is fed into the network as one batch (training proceeds one forward and one backward propagation). Since all the training example starts from a similar metrological state (river flow at a base flow without precipitation), it is possible to set the initial hidden and cell state to be learnable parameters. At prediction time, the input sequence can either (1) start from a base flow state using the learned initial hidden and cell state, or (2) start at the middle of an event using a hot-start hidden and cell state from the prediction of the first part of the event. From a grid search of hyperparameters, a two-layer LSTM network with 64 hidden LSTM units and zero dropout at each layer has been chosen.

3.2.2 Sequence input single output (SISO) model

For the sequence input single output model, we tested four different window size (memory length): 48 steps, 96 steps, 192 steps, and 672 steps. These choices correspond to 12 h, one day, two days, and a week, respectively. The progressive choices are designed to show the cascade of long term dependency of RR. The tuned hyperparameters of the SISO models are listed in Table 2.

For all LSTM networks used in this study, a fully connected layer with ReLU activation is added after the last LSTM layer to map the high dimensional hidden state vector h_t at each time step t to a scalar output Q_t . All LSTM networks are trained with Adam optimizer [30] with the AMSGrad variant [31] and the learning rate is set to 0.00005. The models are implemented using deep learning framework PyTorch [32]. Numerical experiments were conducted on RTX node of Frontera at the Texas Advanced Computing Center (TACC). Jupyter notebook hosted on designsafe [33] was used for post-processing and result analysis.

It should be noted that the batch size, early stopping epoch, and the maximum epoch number are not tuned to avoid overfitting. The batch size is set to satisfy the memory requirement of GPU. Early stopping epoch and maximum epoch numbers are set high enough so that the optimization algorithm converges within the limit. The

early stopping round and maximum epoch for SISO168 is set lower than other SISO models with shorter window size since training the SISO168 is significantly more time-consuming. We have to set it to lower so that training does not exceed the 24-h time limit for computing jobs in Frontera.

3.3 Evaluation metrics

Extensive evaluation was conducted using both classic and hydrologically relevant metrics. We used Root Mean Square Error (RMSE), Nash–Sutcliffe Efficiency coefficient (NSE), Mean Absolute Error (MAE) to measure the model performance. While RMSE could provide valuable information on the model performance, breaking it down to bias, amplitude error, and the phase error could provide more specific details on the source of error. The following equations express the RMSE decomposition [34]:

$$RMSE = \sqrt{\frac{\sum_{k=0}^N (Q_{M,k} - Q_{O,k})^2}{N}} \tag{7}$$

$$RMSE^2 = bias^2 + SD_{bias}^2 + DISP^2 \tag{8}$$

$$bias = \frac{1}{N} \sum_{k=0}^N (Q_{M,k} - Q_{O,k}) \tag{9}$$

$$SD_{bias} = \sigma_M - \sigma_O \tag{10}$$

$$DISP = \sqrt{2\sigma_M\sigma_{Obs}(1 - \rho_{O,M})} \tag{11}$$

where $Q_{M,k}$ and $Q_{O,k}$ are the modeled and observed discharges at time step k , respectively; N is the total number of time steps; SD_{bias} is the amplitude error; $DISP$ is the absolute value of phase error where a non zero value indicates the phase of modeled discharge lags or leads the observed one; σ_M and σ_O are the standard deviation of modeled and observed discharges, respectively; and $\rho_{Obs,M}$ is the correlation coefficient between the observed and modeled discharges.

To exploit the hydrological context, hydrologically relevant metrics that evaluate overall water balance, vertical redistribution, and temporal redistribution could be used as diagnostic tools. Some of these metrics are derived from the concept of the flow duration curve. Flow duration curve

Table 2 Tuned hyperparameters of LSTM models

Test name ^a	Hidden size	Drop out	Windows size	Batch size	Early stopping	Max epoch
SISO12	64	0.5	48	4096	400	3000
SISO24	64	0.1	96	1024	400	3000
SISO48	128	0.3	192	1024	400	3000
SISO168	128	0	672	1024	200	1000

^aSISO synced sequence input and output, SISO sequence input and single output, numbers after the SISO show the length of window size in hours

(FDC) is defined as the relationship between a given discharge value and the percentage of time that this value was exceeded. The concept of probability distributions cannot be applied in RR process due to the existing correlation among discharges in successive time and the effect of seasonality. FDC could be considered as the complement of the cumulative distribution function [35]. In this paper, the following hydrologically relevant metrics were used:

$$FMS = \frac{[\log Q_{M,0.2} - \log Q_{M,0.7}] - [\log Q_{O,0.2} - \log Q_{O,0.7}]}{[\log Q_{O,0.2} - \log Q_{O,0.7}]} \tag{12}$$

$$FHV = \frac{\sum_{i=0}^M (Q_{M,i} - Q_{O,i})}{\sum_{i=0}^M Q_{O,i}} \tag{13}$$

$$FLV = \frac{\sum_{j=1}^H [\log Q_{M,j} - \log Q_{M,Min}] - \sum_{j=1}^H [\log Q_{O,j} - \log Q_{O,Min}]}{\sum_{j=1}^H [\log Q_{O,j} - \log Q_{O,Min}]} \tag{14}$$

$$FMM = \frac{\log Q_{M,Median} - \log Q_{Ob,Median}}{\log Q_{O,Median}} \tag{15}$$

$$EQ_{Peak} = \frac{(Q_{Peak,M} - Q_{Peak,O})}{Q_{Peak,O}} \tag{16}$$

$$ET_{Peak} = |T_{Peak,M} - T_{Peak,Obs}| \tag{17}$$

where FMS: bias in flow FDC midsegment slope which evaluates the vertical redistribution, $Q_{0.2}$ and $Q_{0.7}$ are discharges associated with the exceedance probabilities of 20% and 70%, FHV is the bias in FDC high-segment volume (2%), M is the number of runoff indices corresponding to discharges with exceedance probabilities smaller than 2%, FLV is the bias in FDC low-segment volume that evaluates the long-term baseflow, H is the number of runoff indices corresponding to discharges with exceedance probabilities smaller between 70 and 100%, FMM is the bias in the median runoff, EQ_{Peak} is the error of peak runoff, and ET_{Peak} is the error of time to peak runoff.

4 Results

The prediction performance is shown in Table 3. Since the test dataset has an unprecedented event, Hurricane Harvey, where flooding was so severe that inter-basin flow was observed, we reported both the evaluation metrics of the entire test set and the ones excluding Harvey.

Among all the methods we tested, the synced sequence input and output model (SSIO) has the best overall performance. It leads all the methods in terms of NSE, MAE, RMSE, and $\rho_{O,M}$. As the length of time window increases,

better overall prediction accuracy is observed for the sequence input and single output models (SISO). NSE, MAE, RMSE, and $\rho_{O,M}$ show a clear progressive increase as the time window increases from 12 to 168 h. This phenomenon suggests that the runoff at the studied watershed indeed has a long term dependency on past rainfall history; thus, architectures that could preserve long term memory are required to model flood on a fine time scale. Phase error (DISP) was the most significant contributor to the RMSE, followed by amplitude error. For large events, the developed model had a lead in predicting the peak flow, which is probably the reason behind the large DISP values. Table 3 also shows SSIO has the best prediction performance in terms of hydrological related evaluation metrics. SSIO has the best bias in median runoff FMM among all the methods. The better performance of the SSIO method in predicting the median discharge could also be seen in Fig. 4a.

Figure 4b depicts the FDC for all test cases. The vertical and horizontal axes show the exceedance rates and their corresponding discharge rates (cfs), respectively. An exceedance rate for a specific discharge rate means what percents of all flows have a value greater than that rate. Within the FDC (Fig. 4b) low-segment region (70–100%), the SSIO model also shows superior performance. It has significantly lower FLV compared to the other methods. Note that for this metric excluding Hurricane Harvey does not change the result significantly, because most of the data points in Hurricane Harvey are outside this region due to large discharge rates during this event. We believe the SSIO model benefits from setting the initial hidden state to be learnable during training. The setting essentially forces the network to learn the base flow condition from data.

Figure 4a also provides a better illustration of the FDC mid-segment region (corresponding bias: FMS) by showing the 25, 50, and 75 percentiles of discharge values for all test datasets. Here again, SSIO showed the most similar pattern to the observed data, followed by SISO168 and 48. Within FDC high-segment region, SSIO has similar FHV with SISO48 and SISO168 on the entire dataset. If we exclude Harvey, the SSIO has the smallest absolute value of FHV. This suggests its superior prediction performance within the high flow region, which is possibly due to its ability to preserve the entire memory of runoff events with a long duration. In flood risk management, we are particularly interested in predicting the magnitude and the time of the peak flow for runoff events. Evaluation metrics for two demonstrative events are shown in Table 4. In addition to Hurricane Harvey, we also looked at the Tax Day Flood event (2016), as the second-largest event in the history of Brays Bayou, which is part of the validation dataset. This particular storm was chosen because the peak flow on this

Table 3 LSTM models prediction performance using the test dataset

Test name	Data set	NSE	MAE	RMSE	SD _{bias}	$\rho_{O,M}$	DISP	Bias	FMS (%)	FHV (%)	FLV (%)	FMM (%)
SSIO	Test	0.943	132.7	614.9	421.0	0.982	443.4	- 65.32	- 21.1	- 16.5	- 64.7	2.6
	Test no Harvey ^a	0.936	72.0	242.0	- 47.8	0.971	237.1	- 5.42	- 20.2	- 0.8	- 67.0	2.4
SISO 12	Test	0.803	250.0	1145.0	768.1	0.923	847.4	- 54.96	- 88.0	- 30.3	- 250.9	- 9.8
	Test no Harvey	0.887	146.9	322.7	- 20.4	0.946	318.5	47.28	- 87.3	- 5.0	- 256.0	- 10.1
SISO 24	Test	0.898	191.5	822.4	541.6	0.964	617.3	- 43.93	- 81.0	- 21.6	- 198.7	- 6.7
	Test no Harvey	0.921	114.9	269.5	- 25.2	0.962	266.4	32.49	- 80.9	- 3.1	- 203.0	- 7.0
SISO 48	Test	0.928	163.0	694.1	445.1	0.974	531.3	- 38.48	- 48.9	- 15.9	- 128.4	- 2.6
	Test no Harvey	0.914	96.2	281.5	- 80.4	0.964	268.9	22.71	- 50.1	1.4	- 131.6	- 2.8
SISO 168	Test	0.935	154.9	655.5	362.1	0.974	545.8	- 25.36	- 52.0	- 16.3	- 186.0	- 4.9
	Test no Harvey	0.934	96.0	245.8	- 11.6	0.968	243.7	30.08	- 52.5	- 6.2	- 190.2	- 5.2

^aSince Hurricane Harvey was an unseen event, the test dataset was also reported without this event

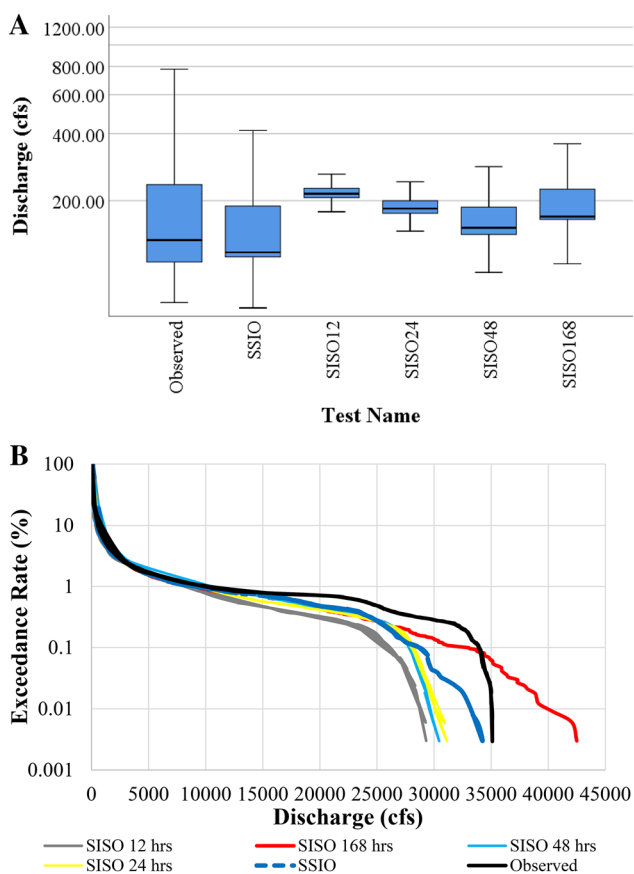


Fig. 4 a Box plot (outliers are not shown) and b flow duration curve (FFC) of test cases for all tested architectures and the observed discharge

event was very similar to Harvey. However, the duration and RR behavior of Harvey were drastically different from Tax Day Flood or any other historical events.

Even though the 2016 event is in the validation dataset and used for hyperparameter tuning, we can still see the superior performance of the SSIO method on FDC high-segment regions. More importantly, from the prediction

evaluation metrics (see Table 3; Fig. 4), we can conclude that the developed LSTM models could predict a very big event (with flow rates as high as 27,200 cfs) with acceptable error. Thus, the evaluation metrics show that the data-driven method can precisely predict runoff of a historic rainfall event. From the NSE/MAE/RMSE metrics in Table 4, we can conclude the SSIO and SISO168 model have similar overall prediction performance on both the Tax Day Flood and Harvey. However, on Harvey, SSIO underpredicted the peak flow by 2.5%, while SISO168 overpredicted it by 21.1%. Here again, it should be noted that Hurricane Harvey was an unseen event in history (flow rates as high as 35,000 cfs), so the models have to extrapolate to predict its discharge rates. Figure 5 shows the LSTM model predictions for Harvey compared to the observed values. From ET_{Peak} and Fig. 5, we can see that all the methods were able to identify the correct pattern of the disastrous runoff event (except SISO 12). But once again, we observed progressively improved performance as the lengths of input memory increases. As the time window size of SISO reaches a week, prediction performance becomes similar to the SSIO model.

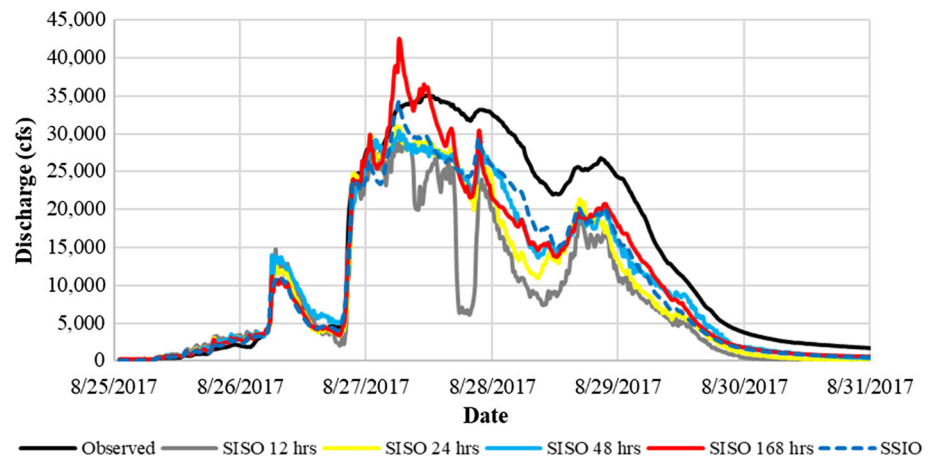
To evaluate the LSTM models' computational time and costs the training times and stopping epochs were recorded. It is found that for SISO architectures with short time windows (12 h or one day), training of one epoch takes a shorter time (3 s and 8 s, respectively) compared to the SSIO model (21 s). As the window size increases to two days and a week, the training time per epoch (25 s and 61 s for 48 and 168 h, respectively) will be longer than the SSIO model. Note that this is not a strict performance timing test, and training time per epoch depends on the number of parameters of the network, batch size, and training strategy. However, the trend of SISO training time suggests its inefficiency. We believe that the training process of SISO architecture is more scalable since it allows larger batch sizes (this batch size is still small enough).

Table 4 LSTM models prediction performance for historically large events in Brays Bayou

Test	Data set	NSE	MAE	RMSE	ET _{Peak}	EQ _{Peak} (%)
SSIO	2016 Event ^a	0.988	436.1	668.5	0.5	1.8
	Harvey ^b	0.922	1891.6	3105.7	5.25	− 2.5
SISO 12	2016 Event	0.885	1141.1	2111.2	0.75	5.6
	Harvey	0.708	3239.1	6026.3	4	− 16.5
SISO 24	2016 Event	0.973	724.9	1020.7	0.5	3.3
	Harvey	0.854	2415.0	4264.3	5.75	− 11.3
SISO 48	2016 Event	0.987	464.7	707.2	2	3.4
	Harvey	0.902	2101.5	3486.7	5.25	− 13.3
SISO 168	2016 Event	0.988	434.5	694.9	4.25	− 2.3
	Harvey	0.910	1863.9	3337.6	5.5	21.1

^aTax Day Flood (2016) was the largest event in the validation dataset

^bHurricane Harvey (2017) was the largest event in the test dataset

Fig. 5 LSTM model predictions for Hurricane Harvey compared to the observed values

However, as the length of time window increases, the extra computation required is going to offset the benefit of scalability at some point. Particularly, during the training of SISO168, we had to reduce the early stopping and maximum training epochs so that the training could be completed within the 24 h limits on TACC. Moreover, the longer time window means a larger memory requirement, which restricts the training batch size. For example, setting the batch size to 4096 for SISO168 will exceed the GPU memory limit.

5 Conclusion

In this study, we have shown the superiority of synced sequence input and output (SSIO) LSTM architecture for hydrologic analysis over existing methods that use sequence input single output (SISO) architecture. The experimental results from real-world hydrologic data validated that SSIO architecture is not only more accurate but also consumes less computational resources. The

advantage of SSIO architecture is especially significant under scenarios where fine temporal resolution is required. Hydrological related evaluation metrics show that the SSIO method has better performance from the hydrologic perspective. Moreover, our results show that particular rainfall events could affect the runoff process in the test watershed for at least a week. Thus, careful treatment of the long term dependency is necessary for fine temporal resolution RR modeling. To this end, we suggest that SSIO is a more suitable architecture for RNN-based RR modeling.

Acknowledgements This research was funded by the National Oceanic and Atmospheric Administration (NOAA, Grant Number NA18NOS0120158), National Science Foundation (NSF, CMMI-1520817), XSEDE Grant NSF-DMS080016N, and generous support from the Texas Advanced Computing Center.

Compliance with ethical standards

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

References

- NOAA National Centers for Environmental Information (NCEI) (2018) U.S. billion-dollar weather and climate disasters. <https://www.ncdc.noaa.gov/billions/>. Accessed 20 Aug 2008
- Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall–runoff models. *Water Resour Res.* <https://doi.org/10.1029/91WR02985>
- Pappenberger F, Beven KJ, Hunter NM et al (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.* <https://doi.org/10.5194/hess-9-381-2005>
- Lili W, Hongjun B, Yu S, Zhongbo Y (2008) Rainfall–runoff simulation and flood forecasting for Huaihe Basin. *Water Sci Eng* 1:24–35
- 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* 6:53–61
- Bedient PB, Holder A, Benavides JA, Vieux BE (2003) Radar-based flood warning system applied to tropical storm Allison. *J Hydrol Eng.* [https://doi.org/10.1061/\(ASCE\)1084-0699\(2003\)8:6\(308\)](https://doi.org/10.1061/(ASCE)1084-0699(2003)8:6(308))
- Downer CW, Ogden FL (2004) GSSHA: model to simulate diverse stream flow producing processes. *J Hydrol Eng.* [https://doi.org/10.1061/\(ASCE\)1084-0699\(2004\)9:3\(161\)](https://doi.org/10.1061/(ASCE)1084-0699(2004)9:3(161))
- A.C.E. US (2010) HEC-RAS river analysis system. User's manual, version 41. <https://www.hec.usace.army.mil/software/hec-ras/documentation/HEC-RAS%205.0%20Reference%20Manual.pdf>
- Dawson CW, Wilby R (1998) An artificial neural network approach to rainfall–runoff modelling. *Hydrol Sci J.* <https://doi.org/10.1080/02626669809492102>
- Young CC, Liu WC, Wu MC (2017) A physically based and machine learning hybrid approach for accurate rainfall–runoff modeling during extreme typhoon events. *Appl Soft Comput J.* <https://doi.org/10.1016/j.asoc.2016.12.052>
- Sudheer KP, Gosain AK, Ramasastri KS (2002) A data-driven algorithm for constructing artificial neural network rainfall–runoff models. *Hydrol Process.* <https://doi.org/10.1002/hyp.554>
- Hettiarachchi P, Hall MJ, Minns AW (2005) The extrapolation of artificial neural networks for the modelling of rainfall–runoff relationships. *J Hydroinform.* <https://doi.org/10.2166/hydro.2005.0025>
- Srinivasulu S, Jain A (2006) A comparative analysis of training methods for artificial neural network rainfall–runoff models. *Appl Soft Comput J.* <https://doi.org/10.1016/j.asoc.2005.02.002>
- 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.* <https://doi.org/10.1080/02626667.2014.967696>
- Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncertainty Fuzziness Knowl Based Syst.* <https://doi.org/10.1142/S0218488598000094>
- Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. *Neural Comput.* <https://doi.org/10.1162/089976600300015015>
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Tian Y, Xu YP, Yang Z et al (2018) Integration of a parsimonious hydrological model with recurrent neural networks for improved streamflow forecasting. *Water* (Switzerland). <https://doi.org/10.3390/w10111655>
- Mhammedi Z, Hellicar A, Rahman A et al (2016) Recurrent neural networks for one day ahead prediction of stream flow. In: *ACM international conference proceeding series*
- Kratzert F, Klotz D, Brenner C et al (2018) Rainfall–runoff modelling using long short-term memory (LSTM) networks. *Hydrol Earth Syst Sci.* <https://doi.org/10.5194/hess-22-6005-2018>
- Wu Y, Liu Z, Xu W et al (2018) Context-aware attention LSTM network for flood prediction. In: *Proceedings—international conference on pattern recognition*
- Yuan X, Chen C, Lei X et al (2018) Monthly runoff forecasting based on LSTM–ALO model. *Stoch Environ Res Risk Assess.* <https://doi.org/10.1007/s00477-018-1560-y>
- Le X-H, Ho VH, Lee G, Jung S (2019) Application of long short-term memory (LSTM) neural network for flood forecasting. *Water* 11:1387
- Kratzert F, Klotz D, Shalev G et al (2019) Benchmarking a catchment-aware long short-term memory network (LSTM) for large-scale hydrological modeling. *Hydrol Earth Syst Sci Discuss.* <https://doi.org/10.5194/hess-2019-368>
- Hu C, Wu Q, Li H et al (2018) Deep learning with a long short-term memory networks approach for rainfall–runoff simulation. *Water* 10:1543
- Kao I-F, Zhou Y, Chang L-C, Chang F-J (2020) Exploring a long short-term memory based encoder–decoder framework for multi-step-ahead flood forecasting. *J Hydrol.* <https://doi.org/10.1016/j.jhydrol.2020.124631>
- Widiasari IR, Nugoho LE, Widyawan, Efendi R (2018) Context-based hydrology time series data for a flood prediction model using LSTM. In: *Proceedings—2018 5th international conference on information technology, computer and electrical engineering, ICITACEE 2018*
- NOAA (2009) NOAA water level and meteorological data report: Hurricane Ike. Maryland
- Li W, Kiaghadi A, Dawson CN (2020) High temporal resolution rainfall runoff modelling using long-short-term-memory (LSTM) networks. *Neural Comput Appl.* <https://doi.org/10.1007/s00521-020-05010-6>
- Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Phuong TT, Trieu Phong L (2019) On the convergence proof of AMSGrad and a new version. [arXiv:1904.03590](https://arxiv.org/abs/1904.03590)
- Paszke A, Chanan G, Lin Z et al (2017) Automatic differentiation in PyTorch. In: *31st conference on neural information processing systems*
- Rathje EM, Clint D, Padgett JE et al (2017) DesignSafe: new cyberinfrastructure for natural hazards engineering. *Nat Hazards Rev* 18:6017001. [https://doi.org/10.1061/\(ASCE\)NH.1527-6996.0000246](https://doi.org/10.1061/(ASCE)NH.1527-6996.0000246)
- De Giorgi MG, Campilongo S, Ficarella A, Congedo PM (2014) Comparison between wind power prediction models based on wavelet decomposition with least-squares support vector machine (LS-SVM) and artificial neural network (ANN). *Energies* 7:5251–5272. <https://doi.org/10.3390/en7085251>
- Iacobellis V (2008) Probabilistic model for the estimation of T year flow duration curves. *Water Resour Res.* <https://doi.org/10.1029/2006WR005400>

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