AGU100 ADVANCING EARTH AND SPACE SCIENCE

Water Resources Research

RESEARCH ARTICLE
10.1029/2019WR025326

Special Section:
Big Data & Machine Learning in Water Sciences: Recent Progress and Their Use in Advancing Science

A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning

Zhongrun Xiang[1], **Jun Yan**[2], **and Ibrahim Demir**[1]

[1]Department of Civil and Environmental Engineering, University of Iowa, Iowa City, IA, USA, [2]DHI China, Shanghai, China

Abstract Rainfall-runoff modeling is a complex nonlinear time series problem. While there is still room for improvement, researchers have been developing physical and machine learning models for decades to predict runoff using rainfall data sets. With the advancement of computational hardware resources and algorithms, deep learning methods such as the long short-term memory (LSTM) model and sequence-to-sequence (seq2seq) modeling have shown a good deal of promise in dealing with time series problems by considering long-term dependencies and multiple outputs. This study presents an application of a prediction model based on LSTM and the seq2seq structure to estimate hourly rainfall-runoff. Focusing on two Midwestern watersheds, namely, Clear Creek and Upper Wapsipinicon River in Iowa, these models were used to predict hourly runoff for a 24-hr period using rainfall observation, rainfall forecast, runoff observation, and empirical monthly evapotranspiration data from all stations in these two watersheds. The models were evaluated using the Nash-Sutcliffe efficiency coefficient, the correlation coefficient, statistical bias, and the normalized root-mean-square error. The results show that the LSTM-seq2seq model outperforms linear regression, Lasso regression, Ridge regression, support vector regression, Gaussian processes regression, and LSTM in all stations from these two watersheds. The LSTM-seq2seq model shows sufficient predictive power and could be used to improve forecast accuracy in short-term flood forecast applications. In addition, the seq2seq method was demonstrated to be an effective method for time series predictions in hydrology.

Key Points:
- An hourly runoff model was developed using the LSTM sequence-to-sequence learning method for 24-hr predictions on USGS stations
- The proposed model shows better performance than traditional data-driven models and is applicable to different watersheds
- The advantages and limitations of seq2seq models and how this model structure could work on the rainfall-runoff modeling is presented

Supporting Information:
- Supporting Information S1

Correspondence to:
Z. Xiang,
zhongrun-xiang@uiowa.edu

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

Received 16 JUL 2019
Accepted 28 DEC 2019
Accepted article online 3 JAN 2020

1. Introduction

Floods tend to happen during heavy rainfall events, which generate too much surface runoff. Surface runoff is an important driving force of the transport of sediment, nitrate, phosphorus, and other chemical compounds in the watershed (Neitsch et al., 2011). Thus, rainfall-runoff modeling is very important in both hydrological and environmental studies. Many modeling frameworks, information systems, and applications have been developed and are widely used in hydrological studies for the prediction and analysis of flooding (Demir et al., 2018, Demir & Szczepanek, 2017; Krajewski et al., 2017) and water quality (Jones et al., 2018; Weber et al., 2018). These models are powered by extensive data collection and sensor networks (Demir et al., 2015).

Hydrological analysis models include physical models with 1-D, 2-D, and 3-D simulation capabilities. The Hydrological Simulation Program-FORTRAN (HSPF) is software used to simulate the watershed hydrology and water quality using a 1-D basin-based method (Johanson et al., 1996). The Soil and Water Assessment Tool (SWAT) has been integrated with the ArcGIS interface to create a 2-D basin-based hydrology model that is widely used in the runoff and water quality modeling in agricultural areas (Neitsch et al., 2011; Xiang et al., 2018). Mike SHE and Mike3 provide 3-D simulations of surface flow and sediment in urban, coast, and sea areas, and they both require high-quality physical data for calculating physical equations (Devia et al., 2015).

In addition to physically based models, statistical and machine learning methods have been used for runoff predictions as well. Support vector machines (SVMs) are models that use a linear or nonlinear kernel weighting on the input variable to find the smallest output error linearly (Granata et al., 2016). Similar to linear regression, SVMs can model a single time step in a time series while being reported to have better modeling speed and accuracy than physical models such as MIKE Flood (Yan et al., 2018) and Storm Water Management Model (SWMM) (Granata et al., 2016). Artificial neural networks (ANNs) are a type of machine learning methods that have been used for hydrological modeling since the 1990s (Abrahart et al., 2004; Hsu et al., 1995). It was known in early studies as multilayer perceptron, a basic structure with multiple dense layers. In the last few decades, several studies (Chang et al., 2015; Ömer Faruk, 2010; Sajikumar & Thandaveswara, 1999) have reported that the performances of ANN models are comparable to those of physical models, and they work well in areas with limited data (Ghumman et al., 2011). They can run fast because they do not require modeling of the internal structure (Hsu et al., 1995), and they can be more efficient with the standard support of hardware, software, and algorithm parallelization (LeCun et al., 2015). There are many different ANN architectures used for hydrological modeling, such as the backpropagation neural network (Sajikumar & Thandaveswara, 1999) and recurrent neural networks (RNNs) (Nagesh Kumar et al., 2004). Although robust ANN model development approaches are still needed (Maier et al., 2010), studies have shown that ANNs can predict the runoff more accurately than can the classical regression models (Mosavi et al., 2018; Riad et al., 2004).

However, earlier ANN models had a limited number of hidden layers due to algorithmic or computational limitations. In recent years, this problem has been overcome with algorithmic advances and accelerated GPU computing (De Donno et al., 2010). To deal with input sequences, RNNs that include long short-term memory (LSTM) and gated recurrent units have been developed and widely used in many studies (Kumar et al., 2016; Sit & Demir, 2019; Wen et al., 2015). Researchers have been using neural networks with multiple nonlinear hidden layers and techniques such as the dropout method, rectified linear unit (ReLU) function, and mini-batch training method, which have brought a revolution in computer vision fields (LeCun et al., 2015). The sequence-to-sequence (seq2seq) learning structure has been developed based on RNNs to solve sequence tasks, and Sutskever et al. (2014) have produced state-of-the-art language translation results with seq2seq learning. With these new methods, researchers have achieved significant improvements in applications for language translation (Cho et al., 2014), speech recognition (LeCun et al., 2015), the game of Go (LeCun et al., 2015), social media data analysis (Sit et al., 2019), intelligent systems (Sermet & Demir, 2018), and many other fields.

In the past 2 years, studies have applied the LSTM model to soil moisture modeling (Fang et al., 2017), monthly water table depth predictions (Zhang et al., 2018), and daily or hourly rainfall-runoff modeling (Hu et al., 2018; Kratzert et al., 2018). However, these models have not produced multiple-step, continuous predictions. For example, Kratzert et al. (2018) used LSTM to predict runoff for the next day, and Hu et al. (2018) used LSTM to predict the runoff for the next 6 hr.

Recently, Reichstein et al. (2019) proposed that language translation modeling methods could be applied to time series tasks. Given the limitations of current short-range hourly runoff models, a new approach for short-range hourly predictions at high temporal resolution is needed for flood preparedness and response (Demir et al., 2018). Thus, this paper presents a continuous hourly rainfall-runoff model for the next 24 hr that uses new deep learning techniques including LSTM, seq2seq learning, ReLU activation function, dropout regularization, mini-batch training, and GPU acceleration. Other machine learning methods including multivariate linear regressions, SVMs, Gaussian process regression (GPR), and regular LSTM are used to evaluate the proposed model performance.

The following section introduces the LSTM model and LSTM-based seq2seq structure, model design, settings and parameterization, and benchmark methods. Section 2.3.1 describes the case study, model optimization, results, and discussion. The conclusion provides a summary of the work and recommendations for future research.

2. Methods

2.1. LSTM-Based seq2seq Basics

LSTM is a special type of RNN with memory structures for learning long-term information. Traditional ANNs cannot connect previous information to the current time step when dealing with long-term dependencies, and LSTM was designed to handle this issue (Hochreiter & Schmidhuber, 1997; Olah, 2015). The basic structure of the modern LSTM model and the algorithms in the cells are shown in Figure 1. LSTM contains a specific status called the cell state C for each time step, which contains the information for long-term memory. The time series input is presented as x in the bottom, and the output is presented as h on the top of Figure 1.

Each LSTM cell updates six parameters in each time step. The detailed algorithms are shown in equations (1) to (6). The first parameter for each cell is the forget gate parameter f_t, which decides how

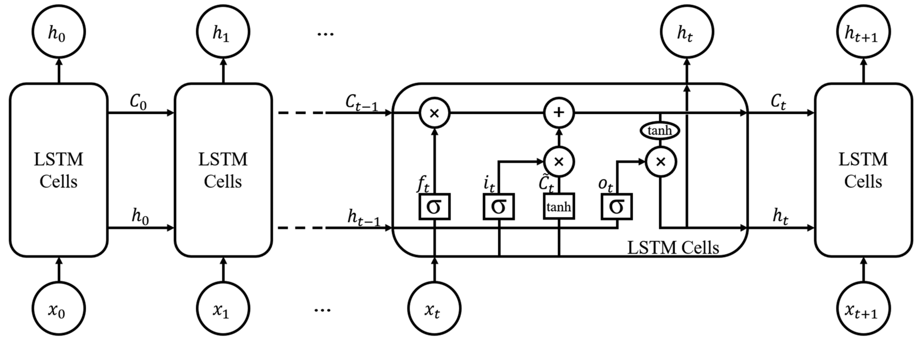

Figure 1. Basic LSTM layer structure for the time step 0 to $t + 1$, with a detailed calculation illustration shown in the LSTM cell at time step t.

much of the previous cell state needs to be forgotten by a sigmoid function with a linear calculation on the current input x_t and the previous result h_{t-1}. The linear equations in different steps have different weights (W) and biases (b) in each LSTM cell. The closer ft is to 0, the more the sigmoid function forgets the previous cell state C_{t-1}. The second parameter is the input gate, which decides what new information is going to be remembered by adding it to the cell state. The input gate parameter i_t is calculated by the sigmoid function with a linear relation on x_t and h_{t-1} as well. $\widetilde{C}_t$ is the candidate of new cell state values, and it was calculated by a tanh function with a linear relation on x_t and h_{t-1}. Then, the cell state C_t is updated. In the end, the output parameter o_t is calculated by the sigmoid function with a linear relation on x_t and h_{t-1}. The final output result at the current step, h_t, is the production of o_t and the tanh function value of cell state C_t.

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

$$i_t = \sigma(W_i\cdot[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = \tanh(W_C\cdot[h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t\times C_{t-1} + i_t\times\widetilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o\cdot[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t\times\tanh(C_t) \tag{6}$$

In this study, due to the time cost of the physical processes converting rainfall to runoff and the time of concentration of the watershed, the runoff at stations would be depending on the rainfall in the past dozens of hours. LSTM is selected as the neural network algorithm used to deal with time series data such as rainfall and runoff observations. The LSTM layer is available as a standard package in many modern machine learning software packages (e.g., Tensorflow and Keras). Thus, this project utilized the built-in LSTM layer components in the Keras framework.

LSTM can solve the problem of long-term dependencies, but it has the limitation of requiring the same time steps for the input and output as shown in Figure 1. However, in many cases, such as the prediction of runoff in this project, it is necessary to know the rainfall at previous hours other than the rainfall at the hour we need to predict. Thus, Cho et al. (2014) proposed a neural network structure called Encoder-Decoder, or seq2seq, which allows the model to be established on different input and output time steps.

Figure 2 illustrates an LSTM-based seq2seq model structure. The final output from the encoder LSTM with m time steps can be stored in a cell named state vector and then used as input for the decoder LSTM with n time steps. This seq2seq structure has solved the time step issue because the steps of encoder input and decoder output can be different. Thus, the LSTM-based seq2seq structure was selected as the basic structure for this hourly rainfall-runoff modeling study. For the rainfall-runoff task, x_i represents the rainfall and runoff observations; h_m represents the coded vector that contains all the necessary information from x_i; and y_i represents the runoff predictions.

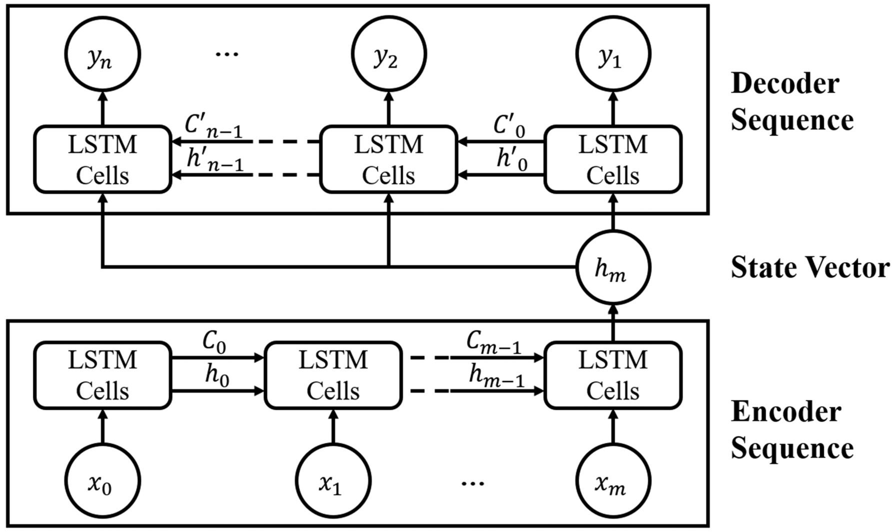

Figure 2. One-layer LSTM-based seq2seq model structure with m time step input and n time step output.

2.2. Model Design

The standard structures of the LSTM-seq2seq model shown in Figure 2 are originally designed for language translation tasks. This study developed an LSTM-based seq2seq model for the hourly runoff prediction of the next 24 hr. In this model structure (Figure 3), time series input, including the rainfall and runoff, is represented separately due to different time steps. Thus, the first LSTM encoder for the runoff observations has a time step of m, and the second LSTM encoder for the observation and forecast data (i.e., rainfall data) have the time step of $m + 24$. Additional time series data, such as the monthly evapotranspiration (ET), were used to be integrated into the state vector as well. The final runoff forecast will be the output after several dense layers. For the downstream stations, the upstream forecast will be helpful for the prediction. Thus, the second LSTM encoder includes the most recent upstream modeling data as well. The overall model can be simplified as

$$\text{Runoff}_{t+23,t+22,\ldots,t} = f\left(\text{Runoff}_{t-1,t-2,\ldots,t-m}, \text{Rainfall\&Upstream}_{t+23,t+22,\ldots,t-m}, \text{ET}_t\right) \quad (7)$$

2.3. Model Settings and Parameterization

The goal of this project is to develop an applicable model to different watersheds rather than to pursue the highest accuracy in the study area. Default settings and common values of the deep learning models were

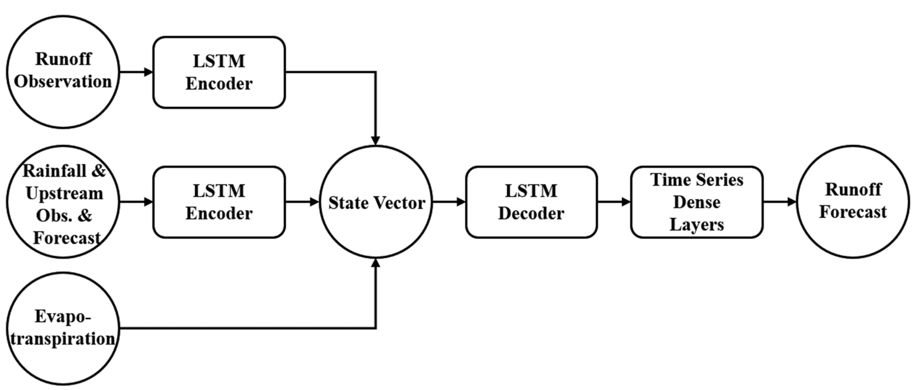

Figure 3. The rainfall-runoff model stated based on the LSTM-based seq2seq model with input of hourly rainfall and runoff observations, ET of the month for time t, and hourly rainfall forecast for a 24-hour runoff prediction.

10.1029/2019WR025326

used in this study as a base model. Three feature-related parameters were evaluated in tuning the model, including input hours of observations, moving average of rainfall data, and batch size.

2.3.1. Model Settings

For each input encoder LSTM layer, 256 neurons were used, and the decoder is one LSTM layer with 512 neurons. After the decoder layer, there are six time series dense layers with neurons ranging from 512 to 1. All dense layers used the ReLU activation function. In past decades, researchers used smoother nonlinearities, such as tanh or sigmoid function; however, the biologically inspired ReLU function [$R(y) = \max(0, y)$] was found to run faster and gain better results. It also helps avoid gradient explosion or disappearance, a problem that makes the weights (W) in equations (1)–(3) and (5) not able to be trained in deep neural networks (LeCun et al., 2015). To avoid overfitting and get a sparse structure, a dropout rate of 0.2 was used to randomly delete 20% of neural connections between nodes during training on dense layers. The optimizer for finding the minimum of error and loss used in this model is RMSprop, a gradient descent-based algorithm that typically works well for RNNs and can progressively approach the minimum of the loss function. The initial learning rate for the RMSprop optimizer is set to 0.0001. One out of three instances of the raw data was extracted as the validation data set, and the learning rate would decrease by a factor of 0.3 when the loss function on the validation data set does not decrease in order to avoid overfitting. The loss function in the model is trying to minimize the fraction of variance unexplained, or the residual sum of squares divided by the total sum of squares (equation (8)). The loss function is also a part of Nash-Sutcliffe efficiency (NSE; equation (9)), which is a widely used performance evaluation method for hydrological modeling (Arnold et al., 2012; Krause et al., 2005). It is also the main evaluation method used in this study.

$$\text{Loss Function} = \frac{\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2} \tag{8}$$

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2} \tag{9}$$

where Y_i is the observation at time i, $\widehat{Y}_i$ is the model result at time i, $\bar{Y}$ is the mean of all observations, and n is the total number of observations.

NSE ranges from $-\infty$ to 1, and the closer its value is to 1, the better the model performs (Arnold et al., 2012). Other performance statistics including Pearson's correlation coefficient (r), percent bias (BIAS), and normalized root-mean-square error (NRMSE) are used for model evaluation (equations (10)–(12)). The defining equations are shown below:

$$r = \frac{\sum_{i=1}^{n}\left(\widehat{Y}_i - \bar{Y}\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2} \tag{10}$$

$$\text{bias} = \frac{\sum_{i=1}^{n}\widehat{Y}_i - \sum_{i=1}^{n}Y_i}{\sum_{i=1}^{n}Y_i} \times 100\% \tag{11}$$

$$\text{NRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2}{n}}}{\bar{Y}} \tag{12}$$

Due to the uncertainties caused by the stochastic processes in deep learning modeling, five duplicate runs were done for each model, and the mean, median, and standard deviation were reported in the results. The independent-sample t test for the model results at the confidence level of 0.95 is used during model development. The models were developed based on Python 3 using the Keras framework with TensorFlow backend, and the NVIDIA Quadro M5000 GPU was used in training.

2.3.2. Parameter 1

Input variables and the time steps are important factors in the LSTM encoders. Due to the limitation of data sources, this project only focuses on three features stated in previous sections, namely, rainfall, runoff, and ET. The forecast time step is fixed at 24 hr in this study based on our goal. Thus, the rainfall and runoff observations from the last 12 to 96 hr were tested, and the 24-hr observed input was set as default in the base model.

2.3.3. Parameter 2

Hourly rainfall data, especially from the short-duration thunderstorms, may cause high variance and noisy signals in the modeling. Nhita et al. (2016) have shown that the moving average of rainfall data is an effective data smoothing method that can improve the modeling and forecasting using neural networks. In this study, simple moving averages (SMAs) with different window size n (equation (13)) of hourly rainfall ranges were tested:

$$Y_t = \frac{1}{2n+1}\sum\nolimits_{i=-n}^{n} X_{t+i} \tag{13}$$

where Y_t is the rainfall after a moving average at hour t and X_t is the hourly rainfall data at time t. The integer n represents the average ranges. The SMA with parameter n (SMAn) ranging from 2 to 10 was tested.

2.3.4. Parameter 3

The batch size, an important parameter in the deep learning model, was tested in this study. Deep learning models select and train instances in batches. There are 43,824 hr in 5 years and ideally 43,824 instances. The model will randomly select two of three instances (29,216) for training and one of three instances (14,608) for validation. The batch size is determined by the project type and normally ranges from 32 to 512. If the batch size is 32 and there are 29,216 instances, it means in each training epoch the model will randomly select and train 32 instances each time without replacement until all instances have been used; the gradient will update 913 times in each epoch. Thus, the larger the batch size, the faster the training is. However, the model quality may significantly decrease when the batch size is too large (Keskar et al., 2016). In this rainfall-runoff project, a suitable batch should contain data from both sunny and rainy hours in both dry and wet seasons. If the batch is a good representation of the whole data set, the model will be well trained and stable. However, if the batch size is too small, the loss function will be hard to converge because each batch is significantly different (i.e., too small batches may contain only dry-hours data). If the batch size is too large, the model may perform well overall but be partially biased. Thus, the default batch size of 512 was used first in the base model, and the values ranging from 32 to 256 were tested.

2.4. Model Benchmarks and Methods

To evaluate the model performance, several benchmarks were used. These benchmarks include runoff persistence, linear least squares regression, Lasso regression, Ridge regression, GPR, support vector regression (SVR) with a linear kernel, SVR with a second-order polynomial kernel, SVR with a radial basis function kernel, and the LSTM model (without seq2seq structure).

The persistence of runoff from 1 to 24 hr was used as a benchmark to evaluate the change in the runoff. It assumes the runoff does not change over time. A high correlation or NSE for persistence represents stable discharge rates, fewer changes, and high autocorrelation over time (Puckridge et al., 2000).

To compare the relationships between input and output, multiple linear regression with the least squares method and regression with L1 regularization (Lasso regression) and L2 regularization (Ridge regression) were used. The multiple linear regression with the least squares method minimized the sum of the squares of the vertical deviations of instances. However, since the input variables (i.e., the observed rainfall and runoff over time) are highly autocorrelated, it may be overfitted. Thus, the L1 or L2 regularization may help to provide a better linear regression for the model. All three linear regressions were tested in this study.

GPR is a nonparametric Bayesian nonlinear regression method that can model random variables with a multivariate normal distribution in continuous time or space (Kirk & Stumpf, 2009). Recent studies have applied GPR to monthly rainfall-runoff predictions for the next time step with acceptable accuracy (W. Chang et al., 2018; Sun et al., 2014). GPR was used as a benchmark in this study as well.

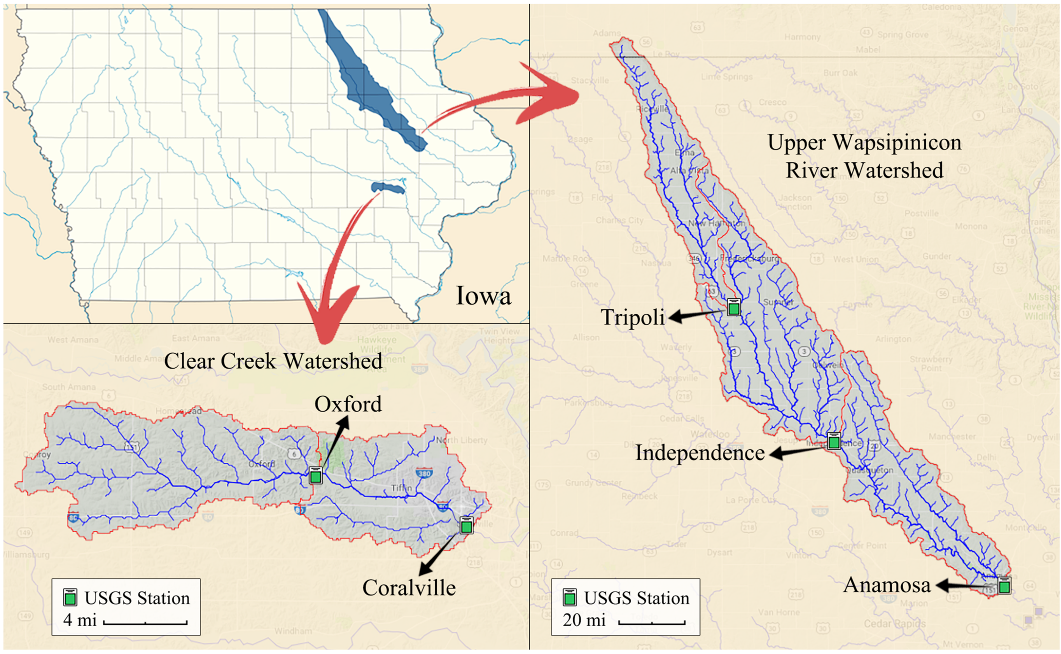

Figure 4. (top left) Two watersheds and their locations in the State of Iowa. (bottom left) The detailed streamflow map of the Clear Creek Watershed and (right) the Upper Wapsipinicon River Watershed with sub-watershed boundary (red line) and USGS stream gauges (green markers).

SVR is a machine learning model that can apply kernel functions to map original data into a higher dimension and to the input for minimizing the ε-insensitive loss function (Awad & Khanna, 2015). Studies have shown that using SVR has great potential in rainfall-runoff predictions (Granata et al., 2016; Yan et al., 2018). The most common kernels are linear, second-order polynomial, and radial basis functions. All three kernels were tested in this study.

Detailed settings of these machine learning methods with preliminary results can be found in the supplemental materials Text S1 and Table S1.

The traditional LSTM model without the seq2seq structure was tested for comparison. All LSTM settings were the same as those for the LSTM-seq2seq model shown in Figure 3 except for the removal of the LSTM decoder layer and the consideration of the time series relationship for the dense layer. Thus, the dense layer directly connects to the combined encoder LSTM output, which is the same as in the models proposed by Hu et al. (2018) and Kratzert et al. (2018).

2.5. Case Study

The first LSTM-seq2seq model was developed for the Clear Creek Watershed located in Johnson County, Iowa. This is a relatively small watershed that covers 66,132 acres (267 km^2) (Iowa Watershed Approach, 2017a). It is well studied because of frequent flooding events in the last few decades, which had a huge impact on the local economy. An additional evaluation was conducted in the Upper Wapsipinicon River Watershed to discover if the proposed model architecture works on a different watershed after a new training process. It is a large watershed that covers 991,980 acres (4,014 km^2) and overlaps with 11 counties in northeast Iowa (Iowa Watershed Approach, 2017b). It has a narrow shape with a much longer time of concentration than has the Clear Creek Watershed. Both watersheds are vulnerable to flooding and part of the Iowa Watershed Approach project (Demir et al., 2015; Weber et al., 2018) for flood risk reduction. Figure 4 demostrates the locations of Clear Creek Watershed and the Upper Wapsipinicon River Watersheds in the State of Iowa.

Two U.S. Geological Survey (USGS) stations have provided discharge information for the past few decades. They are the USGS 05454220 Clear Creek near Oxford (Oxford station) and the USGS 05454300 Clear Creek near Coralville (Coralville station). The Oxford station is the upstream station, and the Coralville station is located at the outlet of the Clear Creek. The time of concentration for the Clear Creek Watershed is

Table 1
NSE, Correlation Coefficient, Bias, and NRMSE for the Runoff Persistence, Lasso, GPR, SVR, LSTM, and the LSTM-Based seq2seq Models in the 24-hr-Ahead Prediction in WY2016 for Coralville Station

Statistical Measures	Persistence	Lasso	SVR	PGR	LSTM	LSTM-seq2seq	LSTM-seq2seq distributed
NSE	−0.348	0.533	0.572	−2.886	0.453	0.649	0.768
r	0.301	0.738	0.768	−0.020	0.722	0.808	0.878
bias (%)	−1.3	−14.7	4.6	5.0	−3.8	−8.7	1.1
NRMSE	1.75	1.03	0.98	2.97	1.11	0.89	0.75

about 24 hr, and the subwatershed of the Oxford station is about 12 hr. There are three USGS stations located in the Upper Wapsipinicon River Watershed: USGS 05420680 Wapsipinicon River near Tripoli (Tripoli station), USGS 05421000 Wapsipinicon River at Independence (Independence station), and USGS 05421740 Wapsipinicon River near Anamosa (Anamosa station). The Anamosa station is located downstream of the Independence station, and the Independence station is located downstream of the Tripoli station.

Available data for the case study includes runoff data from the U.S. Geological Survey (2016), 15-min Radar Stage IV precipitation data (Lin, 2011), and monthly ET data from the Iowa Flood Center (Krajewski et al., 2017). Data from the water years (WYs) 2012 to 2017 were used for modeling. These years include normal, extremely dry, and wet conditions at each watershed. Dry and wet years were selected for training, and a normal year was used for testing. Links to access the data sources are in the supporting information Table S4.

Several preprocessing steps were carried out on the data and model setup. First, rainfall data were aggregated to hourly by summing up, and the runoff data were aggregated to hourly by averaging. Linear interpolation was used for missing data when the gap is less than 12 hr. Min-max scaling was used on all input catalogs to standardize the input, to better initialize the parameter, and to speed up the convergence. The rainfall data obtained from the Radar Stage IV have a high resolution of 0.05°. Thus, to reduce the input dimensions and modeling noise, for each hour, the overall rainfall on the subwatersheds was used as the input rainfall.

The average runoff for the two USGS stations over the WYs from 2012 to 2017 and the average of all 6 years are shown in Table 1. For the Clear Creek Watershed, WY2017, which is a relatively dry year, was the second lowest average runoff in the past 6 years. The NSE for the 24-hr persistence of the Oxford station at WY2017 is 0.50, which means the runoff was very stable in WY2017. However, WY2016 had several flood events with a negative NSE for the 24-hr persistence; thus, WY2016 was selected as the test year for the Clear Creek Watershed. For the Upper Wapsipinicon River Watershed, WY2017 was selected as the test year, and the previous years' observations were used for model calibration. For each station, a total of 5 years was used to calibrate the model. To avoid overfitting, two of the three instances of calibration data were used for training, and the remaining one of the three instances were used for validation. A well-fitted model should have a good accuracy not only in the training data set but also in the validation data set. Final evaluations were all based on the test year, which are independent of the training and validation data sets.

Based on the Oxford station, a base model was developed, and analyses of models with different input hours, moving average ranges of rainfall data, and batch sizes were conducted. Detailed analyses include NSE comparison of model structures, time series plots that compare the LSTM-seq2seq model and all other models, and scatterplots with matrix analysis considering the flood stages. For the Coralville station, stations with and without upstream data were tested.

After the model structure was finalized for the Clear Creek Watershed, it was then applied on three other stations in the Upper Wapsipinicon River Watershed with the calibration of their rainfall and runoff observations.

3. Results

3.1. Model Optimization

3.1.1. Base Model

A base model was set using data from the Oxford station, where the input rainfall and runoff observations are for 24 hr, with a large batch size of 512. The average model NSE of runoff prediction for all 24 hr is

10.1029/2019WR025326

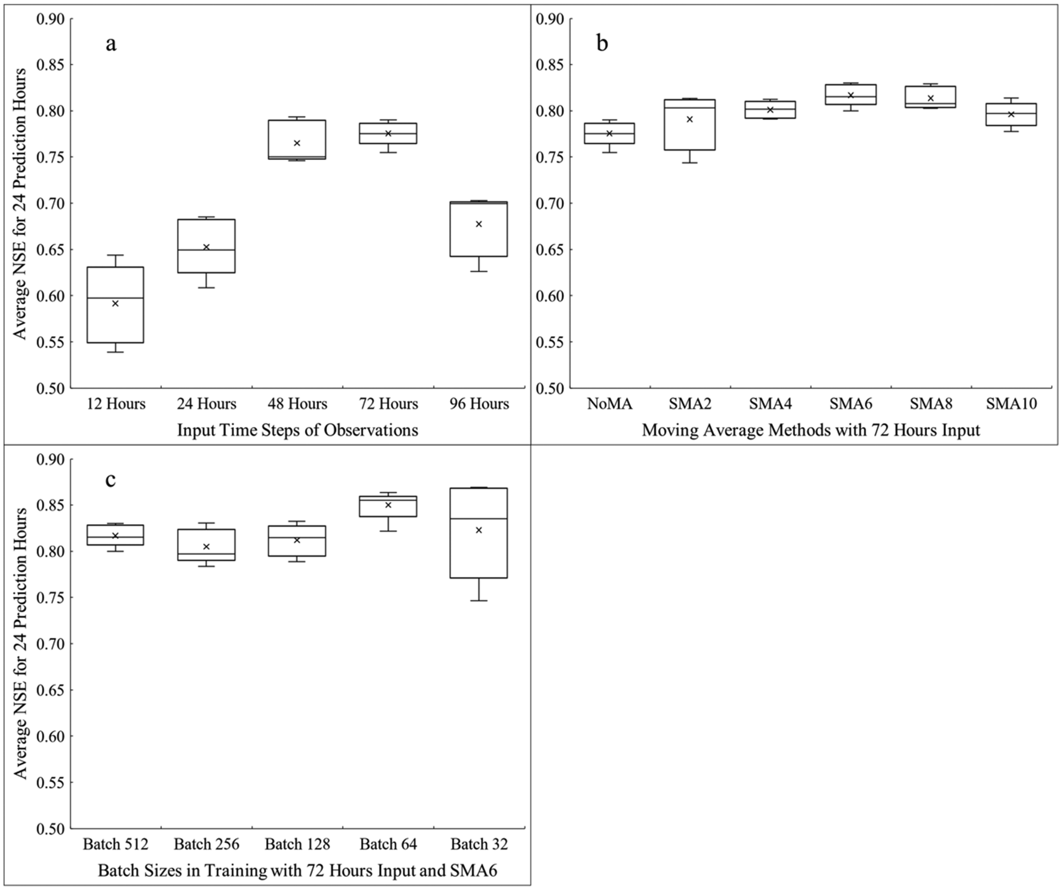

Figure 5. Boxplots of different tests. Each box is calculated from five realizations of model runs. The average NSE for the prediction of all 24 hours with different (a) input time steps m, (b) moving average ranges n, and (c) batch sizes. The × marks inside the box are the average values.

0.653 (Figure 5a). In the hydrologic modeling, NSE > 0.5 is acceptable (D. N. Moriasi et al., 2007; Neitsch et al., 2011), which indicates that a typical seq2seq model without tuning has the potential to predict the runoff with sufficient performance. The following tests show how the fine-tuning of input features and hyperparameters affect the seq2seq model.

3.1.2. Input Time Steps

Feature selection and configuration are important for machine learning model training and validation. However, due to the limitation of data sources, only two time series input parameters hourly rainfall and runoff were used in this project. The observation time steps m for the historical rainfall and runoff were tested from 12 to 96 hr at the Oxford station. The results (Figure 5a) show that the model with 72 hr of observation data has the highest average NSE of 0.775 and the lowest variance. The model with 48 hr of observation data has a slightly lower NSE of 0.765. The models with both more and fewer historical observations show lower accuracy. The model with 96 hr of observations performs worse than does the model with 72 hr of observations, which is possibly caused by the noise created by irrelevant input. The model with 12 hr of observations has the lowest efficiency with an NSE of 0.591, although 12 hr is the time of concentration of the subwatershed for the Oxford station. This indicates that for the prediction of runoff, data with only the time of concentration are not enough. Some factors, such as soil moisture condition, are very important for relatively longer predictions, and they may be influenced by rainfall and runoff from past days. Although the input contains rainfall, runoff, and monthly ET data, factors such as soil moisture may have been represented by this deep learning model indirectly. Based on the mean and median values, the observation of 72 hr was identified to be the best time for the future runoff predictions in this model and was used in the rest of the tests.

10.1029/2019WR025326

3.1.3. Moving Average on Rainfall

SMA methods, with window size *n* ranging from 2 to 10, were applied to the Oxford station rainfall data before the min-max scaling was applied. Results (Figure 5b) of five runs show that the average model efficiency increases from the previous test's baseline of 0.775 to the highest value of 0.817 when the moving average ranges increase, and the efficiency starts to decrease after hitting the limit at SMA6. These results show that the model with SMA6 shows significant higher efficiency than does the model without treatment of moving average rainfall data at a confidence level of 0.95. This data pretreatment method is an important part of modeling, and the SMA for the rainfall helps to reduce noise and increase the model efficiency (Nhita et al., 2016). SMA6 has been identified as the best pretreatment of rainfall for the runoff prediction in this model and used in the rest of the tests.

3.1.4. Batch Size

The hyperparameters in LSTM models can be adjusted to improve the performance. In the seq2seq model structure, the number of LSTM layers and dense layers, the number of neural cells in each layer, and the dropout rate can affect the result. However, this study focuses mostly on the parameters related to input variables. In this analysis, the batch size was adjusted from 32 to 512 and tested at the Oxford station. Results (Figure 5c) show that the average NSE of all 24-hr prediction results is 0.817 with a batch size of 512. There is no improvement when using a batch size of 256 or a batch size of 128. However, when the batch size is decreased to 64 and 32, model efficiency increases to 0.850 and 0.823, respectively. In particular, the model with a batch size of 64 shows significantly higher NSE than does the model with a batch size of 512 at a confidence level of 0.95. These results show that a relatively small batch size may have large variances, which makes it easier to create a more generalized and less overfitted model (Keskar et al., 2016). For the selected study area, each instance contains 72 hr of rainfall observations and 24-hr forecast, which is a range of 4 days. To estimate the coverage, 64 well-spread instances may include information from a maximum of 256 days. This means each batch may include varieties of data (i.e., wet hours, dry hours, rising limb, and falling limb) in the training data set, and this provides sufficient variances for calculating gradient descent in batches. The results with a batch size of 32 show significantly higher variance than do other models, which is caused by insufficient data in a batch. Additional tests also show that the batch size that is smaller than 16 can cause the failure of converging the loss function. It can be concluded from the results that a batch size of 64 is the best representation of the hourly rainfall and runoff information. Thus, the rest of the tests used a batch size of 64.

3.1.5. Impact of ET Input

Traditional seq2seq models include only the sequence input. In the design of this seq2seq model, a nonsequence layer for ET data was developed as a new approach. To evaluate the impact of ET, this nonsequence layer was removed, and all other structures have been kept the same for the Oxford station. The NSEs of the model without ET are 0.962 and 0.894 for 1-hr and 3-hr predictions, respectively, which does not show a significant difference to the NSEs of the model with the ET layer. However, the results decreased significantly for the predictions for over 3 hr. Removing the ET layer caused the NSE for the 6-hr-ahead prediction to decrease from 0.856 to 0.809, and the 24-hr-ahead prediction NSE to decrease from 0.789 to 0.753. The average NSE of all 24 hr decreased from 0.850 to 0.785. This indicates that although the ET data are not an hourly time series input, they still significantly affect the model efficiency for the predictions over 3 hr. This indicates that nonsequence information such as watershed size, elevation information, land cover, and other seasonal data may be included in the seq2seq model to develop a more generalized model for multiple watersheds in the future.

3.2. Understanding LSTM-seq2seq in Hydrology With Model Evaluations

Using the same input, other models were tested for the Oxford station. Among linear models, the Lasso model shows the best accuracy compared to normal linear and Ridge regression models. Among the SVR models, SVR with a radial basis function kernel shows the best results. These model results are shown in Figure 6.

The persistence of the runoff observation is negative after 14 hr, which suggests the Oxford watershed has a strong dependence on the rainfall events in the test year. Within the linear regression models, Lasso regression shows that the NSE varies from 0.53 to 0.20 for prediction hours from 1 to 24. Linear regression is the simplest model, while the relationships between rainfall, runoff, and ET are not simply linear. GPR is the second model tested in this study. It is a Naïve Bayes-based model that provides predictions based on the

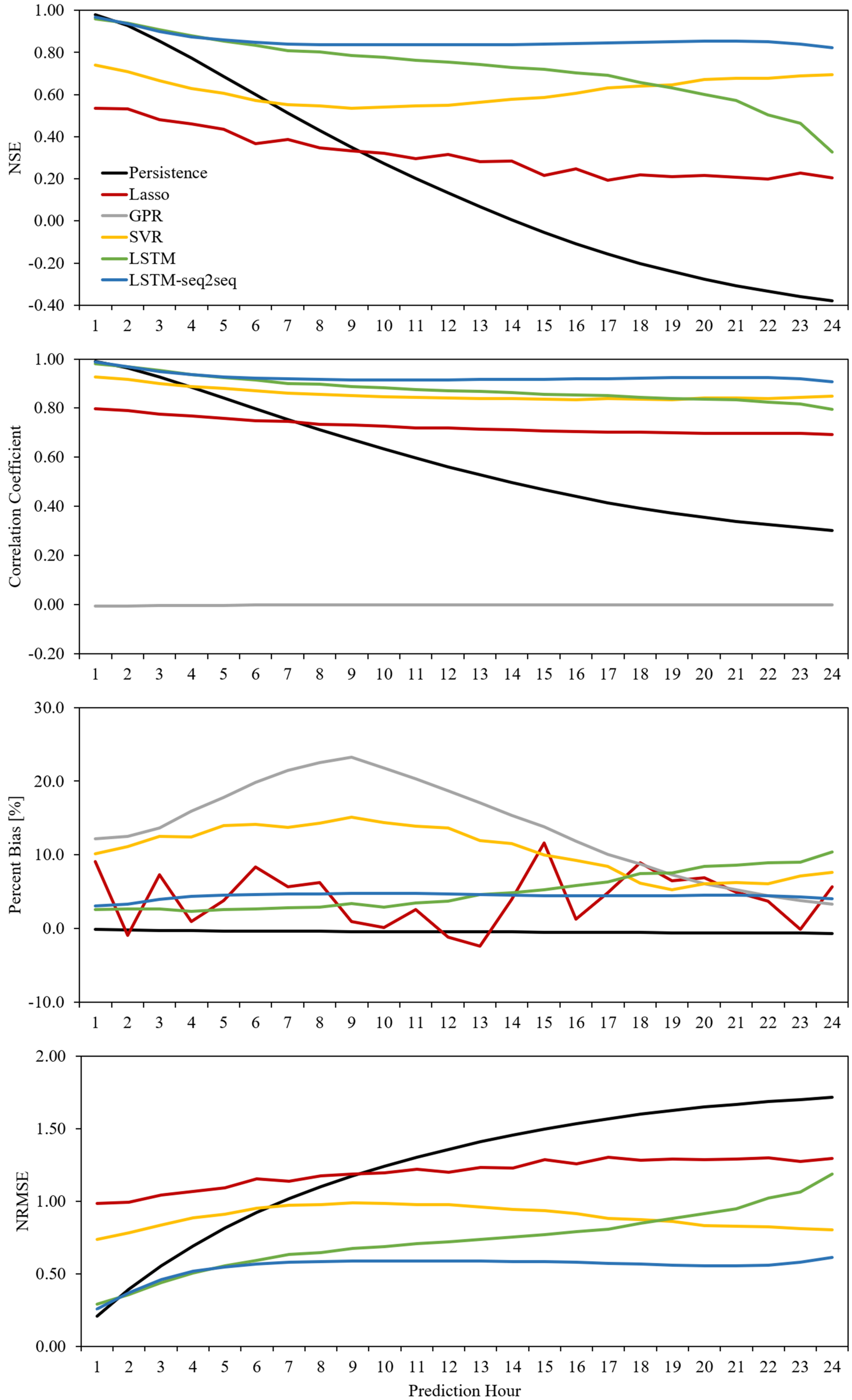

Figure 6. NSE, correlation coefficient, percent bias, and NRMSE for the runoff persistence, Lasso, GPR, SVR, LSTM, and the LSTM-seq2seq model in each of the prediction hours in WY2016 for the Oxford station. NSE and NRMSE for GPR are not shown because the values are extremely bad and not shown on the scales.

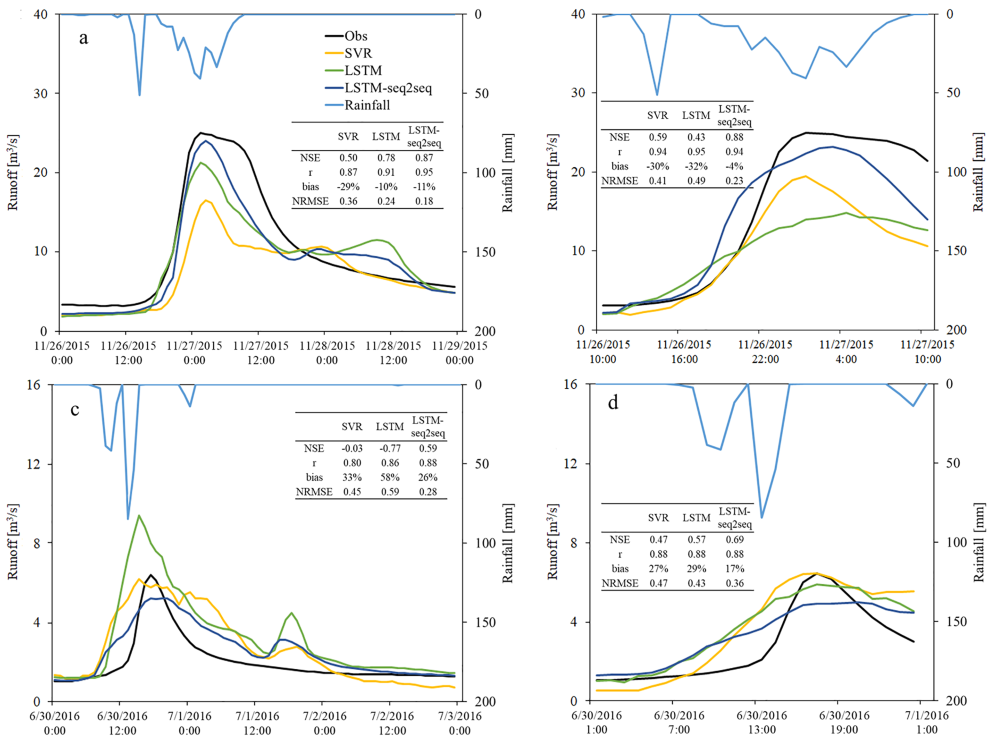

Figure 7. Observations and model predictions of two rainfall events in WY2016. (a, c) The 24-hour-ahead predictions for two events in a 3-day window. (b, d) A single prediction for the next 24 hr on November 26, 2015, 10 a.m., and June 30, 2016, 1 a.m.

previous observations. GPR shows an NSE less than −1 and a correlation coefficient equal to 0, which is not good for any single hourly prediction. This result matches model comparison studies in other fields that have shown that GPR predictions of each data point look plausible but far away from the actual situation (Fragkiadaki et al., 2015). GPR has a bias of 3.3% at the 24-hr-ahead prediction for the test year, which is the smallest among all models and only good for the estimation of the yearly average. SVR with a radial basis function kernel is the best model among SVMs for regression, and it provides a nonlinear regression with the kernel transformations. The NSE for SVR is 0.74 at the 1-hr-ahead prediction, it drops to 0.54 at the 9-hr-ahead prediction, and it goes back to 0.70 at the 24-hr-ahead prediction. SVR is the only model that does not show a monotonic decreasing trend when the prediction hour increases. The reason is that the first several hours' runoff is highly related to the previous runoff, and the runoff over 18 hr is highly related to the rainfall, while the middle parts are relatively hard to predict with SVR. All the models excluding the LSTM and LSTM-seq2seq do not consider the time series relationship among input variables. LSTM treats the hourly input rainfall and runoff in time series, and the NSE and *r* of the model both perform better than SVR in the first 18 hr. However, model performance drops significantly and gains only 0.33 NSE for the 24-hr-ahead prediction. Since the regular LSTM cannot deal with the sequence input and output together, the regular LSTM model only considers the input in time series but predicted the output without the time series relationship (Hu et al., 2018; Kratzert et al., 2018; Zhang et al., 2018). This caused an increasing bias and a significant decrease in the model performance for more time steps. With an additional LSTM decoder layer to treat the output as a sequence, the final LSTM-seq2seq model shows much better results statistically. The NSE ranges from 0.97 to 0.82 for predictions from 1 to 24 hr ahead. The LSTM-seq2seq model performs similarly to the LSTM model in the first 8 hr. After that, it performs better than regular LSTM because the seq2seq structure captures the relationships between time series outputs. The LSTM-seq2seq model also has a very stable bias through leading time and does not increase like the other tested models, which is another contribution using the seq2seq structure.

Looking at the predictions in time series helps understand the seq2seq model. Figure 7 shows two typical rainfall events in the test year with the combination of predictions of 24 hr ahead and the continuous 24-hr predictions made at 18 hr ahead of the peak flow events.

The rainfall event on November 26, 2015 (Figures 7a and 7b), contributes the runoff with an hourly observation of 25.0 m^3/s in the test year, 10.2 m^3/s higher than the flood action level. Figure 7a is a combination of 144 predictions made 24 hr ahead, which provides the detail of the 24-hr-ahead prediction for the whole rainfall event. Figure 7b provides a 24-hr window for continuous 24-hour predictions made 18 hr before the peak flow. The LSTM-seq2seq model predicts the hourly runoff peak flow and time both well from 18 and 24 hr ahead. LSTM and SVR both underpredicted for most of the time. It is also observed from Figure 7b that both LSTM and SVR accumulate to a huge biased total volume in the prediction of the next 24 hr.

The second event, shown in Figures 7c and 7d, has a maximum hourly runoff observation of 6.4 m^3/s, which is a normal rainfall event. Both figures show that the SVR predicted the hourly peak flow the best, while the LSTM-seq2seq model underpredicted the values. However, an interesting thing is that in Figure 7d three models have the same value of correlation coefficient, and all other statistics show that the seq2seq model performs the best. It is easy to find that the total amount of discharge in the predicted period is less biased in the LSTM-seq2seq model than in other models, although it is not good enough when considering the sharpness of the peak—kurtosis. The seq2seq model considers the output as one variable in the time series, which is the main reason why it outperforms the others. In this rainfall-runoff modeling, despite the finding that the model predicts only the runoff, and it cannot do a mass balance as physical models do, the seq2seq structure can help to reduce prediction bias and error by considering the runoff in all prediction hours together. For this reason, even though the LSTM-seq2seq model does not capture the hourly peak flow perfectly for some events, the overall next-day flow rate prediction was more accurate as well.

There are some limitations with the LSTM-seq2seq model. First, as observed from Figure 7, both events contain more than one rainfall peak, but only one runoff peak is observed. However, all the machine learning models predicted a multiple-peak runoff. Although the LSTM and LSTM-seq2seq models perform better than SVR by considering the rainfall input in time series, there is still room to capture the relationships of rainfall-runoff peaks better. Second, this LSTM-seq2seq model structure contains only rainfall, runoff, and ET data. Thus, this model may not perform as well in periods with snow and spring flooding due to limited snow accumulation data in Iowa, although they are not observed for the test watersheds at the selected test year. New input features such as snow cover data and temperature data could be used to solve this problem in other watersheds in the future. Third, the LSTM-seq2seq model treats a watershed as one hydrological unit and does not include any topography data such as land use and slope. It can only make predictions for locations with observed data (i.e., USGS stations). Thus, for prediction in other watersheds where the topography is different, new calibration with their past years' rainfall and runoff data is required. Fourth, due to the watershed being treated as a single hydrology unit, the spatial inequality of rainfall is ignored when using the average rainfall of the watershed. This would cause a larger bias for larger watersheds that more possibly have spatially unequal rainfall distribution. To address this issue, large watersheds with upstream stations can use the station observation as an additional input to better represent upstream rainfall and improve accuracy by reducing errors due to rainfall spatial inequality. This distributed structure for stations with upstream data is tested in the next section.

3.3. Distributed Structure

Coralville is located downstream of Oxford, which can be treated as a single watershed using the total rainfall in the whole watershed area as the rainfall input. However, because the observation and forecast data from the Oxford station are available, a distributed structure can be used to reduce the rainfall spatial inequality error. In this test, the 24-hr-ahead prediction data for the Oxford station were used as additional input in the LSTM encoder layer as shown in Figure 3. Results of the LSTM-seq2seq model structure with and without upstream data are shown in Table 1. If we treat the Clear Creek Watershed at Coralville station as a single watershed, we have the model NSE of 0.649 for the 24-hr-ahead prediction. However, if we treat the watershed as two subwatersheds, using the seq2seq model prediction at 24 hr in the Oxford station as input, and only use the average rainfall in the Coralville subwatershed, the model efficiency increases to 0.768 for the 24-hr-ahead prediction. The distributed structure makes the prediction more accurate for two reasons. First, the average rainfall data for the sub-basin can represent the Coralville station better than can the average rainfall of the whole watershed, which reduces the error caused by the spatial inequality of rainfall data. Second, the distributed structure used the model prediction of upstream runoff, which used the

Table 2
Statistics of LSTM-seq2seq Model and Other Models for the 24-hr Ahead-Prediction on the Upper Wapsipinicon River Watershed Stations in the Water Year 2017

Stations	Statistical Measures	Persistence	Lasso	SVR	LSTM	LSTM-seq2seq
Tripoli	NSE	0.68	0.68	0.77	0.72	0.85
	r	0.84	0.83	0.88	0.91	0.93
	bias (%)	1.4	−6.2	−7.4	4.7	5.1
	NRMSE	0.74	0.74	0.62	0.69	0.50
Independence	NSE	0.76	0.82	0.72	0.80	0.86
	r	0.88	0.91	0.86	0.93	0.94
	bias (%)	0.6	−3.0	−9.3	7.8	−1.3
	NRMSE	0.54	0.46	0.58	0.49	0.42
Anamosa	NSE	0.88	0.82	0.76	0.93	0.93
	r	0.94	0.91	0.89	0.97	0.98
	bias (%)	1.2	−1.0	0.5	6.3	2.2
	NRMSE	0.29	0.35	0.40	0.21	0.21

upstream stream gauge data indirectly. For a large watershed, the runoff at upstream will affect that at downstream significantly, and the relationship is captured by this distributed model.

3.4. Model Application on a Different Watershed

To evaluate the generality of the model structure, the model was applied on three USGS stations in the Upper Wapsipinicon River Watershed with calibrations using their observations. The Anamosa station is the outlet of the watershed. The Independence station is located at the upstream of Anamosa, and the Tripoli station is located at the upstream of Independence. In the distributed model LSTM-seq2seq for stations at Independence and Anamosa, the upstream predictions of Tripoli and Independence were used respectively. After training the model with data from the Upper Wapsipinicon River Watershed for WY2012 to WY2016, the results based on 24-hr-ahead predictions for different models for WY2017 as a test are shown in Table 2. Compared to the Clear Creek Watershed, this is a different watershed with a narrow shape, long travel time, much larger size, and different topography, soil type, and land use. In the test year, the persistence, Lasso, SVR, LSTM, and the LSTM-seq2seq model show NSEs of 0.68, 0.68, 0.77, 0.72, and 0.85, respectively, for the first station, Tripoli. Stations at Independence and Anamosa have a relatively high 24-hr persistence with NSEs of 0.76 and 0.88, respectively, due to the high baseflow, and the final distributed LSTM-seq2seq model outperforms the persistence and other models with NSEs of 0.86 and 0.93. These results are as good as the ones on the Clear Creek Watershed, which confirms that the LSTM-based seq2seq model developed in this study can be used to get accurate predictions on multiple watersheds.

4. Conclusion

seq2seq learning was developed originally for language translation to make translation smoother and more accurate. Due to similarities with language translation tasks, seq2seq models were considered for use in Earth science time series predictions. This paper proposed a seq2seq learning model for rainfall-runoff predictions, which uses one layer of LSTM to deal with the input and output sequence.

This LSTM-seq2seq model considered a station-based watershed as a single hydrology unit and assumed that the complex topography in the watershed was unchanged during the study years. Several domain characteristics are necessary for a successful application. First, the input time step (i.e., 72) representing the past rainfall hours, is considered not related to the time of concentration or lag time. For example, the time of concentration is 12 hr for the Oxford station, but the model with 12 hr of input data does not perform as well as the model with 72 hr of input data. Second, pretreatment such as the moving average of hourly rainfall helps to reduce the noise and increase the model performance. Third, mini-batch training methods help to reduce the bias, and the batch size of 64 is considered as a good representation of the annual water cycle. Too small a batch size will lead to unstable results due to not having enough wet hourly data in one batch. Fourth, the stream relationships between stations were considered, and the final model is semidistributed, which can use the upstream data as an input for the downstream in a large river network. In addition, different from the seq2seq structure, the model also includes a layer for special parameters with different time resolutions, such as the monthly ET. This special design helps to provide additional monthly or seasonal information for pure sequence models and contributes to the performance of the rainfall-runoff prediction.

Five stations in two watersheds were tested in this study. The Clear Creek Watershed with the test year of WY2016 was selected due to its low persistence. The Upper Wapsipinicon River Watershed is a larger watershed with a special topography in Iowa. The proposed LSTM-seq2seq model outperforms the persistence and all other machine learning models including linear regression, Lasso regression, Ridge regression, SVR with linear radial basis function, and quadratic kernels, GPR, and LSTM on all five test stations in two different watersheds.

The reason why this model outperforms other models is the nonlinear algorithms and that the input and output are considered as a two-time-series sequence. Although this model predicts the runoff only, the seq2seq structure can help to reduce prediction bias and error by considering the output sequence together. This works similarly to the physical models and improves the modeling accuracy significantly compared to the traditional LSTM model.

The proposed LSTM-seq2seq model structure contains only rainfall, runoff, and ET data. Thus, this model may not perform as well in periods with snow and spring flooding due to the snow accumulation data in Iowa, although they are not observed for the test watersheds in the test year. New input features such as snow cover data and temperature data could be used to solve this problem in other watersheds in the future. Also, the LSTM-seq2seq model treats a watershed as one hydrological unit and does not include any topography data such as land use and slope. It can only make predictions for locations with observed data. Thus, the model needs new calibration when applied to a different watershed or station. Although this paper proposed a distributed structure to reduce the error caused by spatially unequal rainfall distribution, the spatial inequality of rainfall is still ignored inside each subwatershed. Including topography and other geospatial data sets may provide better results in the future.

Overall, the proposed LSTM-based seq2seq model is applicable to different watersheds in different topographies. This model only uses hourly rainfall observation, rainfall forecast, runoff observation, and monthly ET as the input. Therefore, it is easy to use even for locations with limited physical data. It can serve as a complement or substitute for physically based models, especially for underperforming watersheds and stations. These results also demonstrate the strong potential of applying deep learning methods to other hydrological problems, specifically other time series tasks.

Acknowledgments
The work reported in this study was possible with the support and work of members of the Iowa Flood Center and Civil and Environmental Engineering at the University of Iowa. We thank the agencies who provided the data for streamflow (U.S. Geological Survey, 2016), precipitation (Lin, 2011), monthly ET data set (Krajewski et al., 2017), and USGS gage-based watershed boundary shapefiles in Iowa (Krajewski et al., 2017).

References

Abrahart, R., Kneale, P. E., & See, L. M. (2004). *Neural networks for hydrological modeling* (Vol. 316). London: CRC Press. https://doi.org/10.1201/9780203024119.ch2

Arnold, J.-F. G., Moriasi, D. N., Gassman, P. W., Abbaspour, K. C., White, M. J., Srinivasan, R., et al. (2012). SWAT: Model use, calibration, and validation. *Transactions of the ASABE*, *55*(4), 1549–1559. https://doi.org/10.13031/2013.42263

Awad, M., & Khanna, R. (2015). Support vector regression. In *Efficient Learning Machines* (pp. 67–80). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4302-5990-9_4

Chang, F.-J., Tsai, Y.-H., Chen, P.-A., Coynel, A., & Vachaud, G. (2015). Modeling water quality in an urban river using hydrological factors—Data driven approaches. *Journal of Environmental Management*, *151*, 87–96. https://doi.org/10.1016/J.JENVMAN.2014.12.014

Chang, W., Chen, X., Chang, W., & Chen, X. (2018). Monthly rainfall-runoff modeling at watershed scale: A comparative study of data-driven and theory-driven approaches. *Water*, *10*(9), 1116. https://doi.org/10.3390/w10091116

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. Bengio, Y. (2014). Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. Retrieved from https://arxiv.org/abs/1406.1078

De Donno, D., Esposito, A., Tarricone, L., & Catarinucci, L. (2010). Introduction to GPU computing and CUOA programming: A case study on FOTO. *IEEE Antennas and Propagation Magazine*, *52*(3), 116–122. https://doi.org/10.1109/MAP.2010.5586593

Demir, I., Conover, H., Krajewski, W. F., Seo, B. C., Goska, R., He, Y., et al. (2015). Data-enabled field experiment planning, management, and research using cyberinfrastructure. *Journal of Hydrometeorology*, *16*(3), 1155–1170. https://doi.org/10.1175/JHM-D-14-0163.1

Demir, I., & Szczepanek, R. (2017). Optimization of river network representation data models for web-based systems. *Earth and Space Science*, *4*(6), 336–347. https://doi.org/10.1002/2016EA000224

Demir, I., Yildirim, E., Sermet, Y., & Sit, M. A. (2018). FLOODSS: Iowa flood information system as a generalized flood cyberinfrastructure. *International Journal of River Basin Management*, *16*(3), 393–400. https://doi.org/10.1080/15715124.2017.1411927

Devia, G. K., Ganasri, B. P., & Dwarakish, G. S. (2015). A review on hydrological models. *Aquatic Procedia*, *4*, 1001–1007. https://doi.org/10.1016/j.aqpro.2015.02.126

Fang, K., Shen, C., Kifer, D., & Yang, X. (2017). Prolongation of SMAP to spatiotemporally seamless coverage of continental U.S. using a deep learning neural network. *Geophysical Research Letters*, *44*, 11,030–11,039. https://doi.org/10.1002/2017GL075619

Fragkiadaki, K., Levine, S., Felsen, P., & Malik, J. (2015). Recurrent network models for human dynamics. In 2015 IEEE International Conference on Computer Vision (ICCV) (pp. 4346–4354). IEEE. https://doi.org/10.1109/ICCV.2015.494

Ghumman, A. R., Ghazaw, Y. M., Sohail, A. R., & Watanabe, K. (2011). Runoff forecasting by artificial neural network and conventional model. *Alexandria Engineering Journal*, *50*(4), 345–350. https://doi.org/10.1016/J.AEJ.2012.01.005

Granata, F., Gargano, R., & de Marinis, G. (2016). Support vector regression for rainfall-runoff modeling in urban drainage: A comparison with the EPA's Storm Water Management Model. *Water*, *8*(3), 69. https://doi.org/10.3390/w8030069

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

Hsu, K., Gupta, H. V., & Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research*, *31*(10), 2517–2530. https://doi.org/10.1029/95WR01955

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., & Lou, Z. (2018). Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water*, *10*(11), 1543. https://doi.org/10.3390/w10111543

Iowa Watershed Approach. (2017a). Clear Creek Watershed. Retrieved November 12, 2018, from https://www.iowawatershedapproach.org/watersheds/clear-creek/

Iowa Watershed Approach. (2017b). Upper Wapsipinicon River Watershed. Retrieved November 12, 2018, from https://www.iowawatershedapproach.org/watersheds/upper-wapsipinicon-river/

Johanson, R. C., Bicknell, B. R., Imhoff, J. L., Kittle, J. L., & Donigian, A. S. J. (1996). Hydrological Simulation Program-FORTRAN user's manual for release 11. Retrieved from http://sdi.odu.edu/mbin/hspf/dos/hspf_v11_entirety.pdf

Jones, C. S., Davis, C. A., Drake, C. W., Schilling, K. E., Debionne, S. H., Gilles, D. W., et al. (2018). Iowa statewide stream nitrate load calculated using in situ sensor network. *JAWRA Journal of the American Water Resources Association*, *54*(2), 471–486.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. Retrieved from http://arxiv.org/abs/1609.04836

Kirk, P. D. W., & Stumpf, M. P. H. (2009). Gaussian process regression bootstrapping: Exploring the effects of uncertainty in time course data. *Bioinformatics*, *25*(10), 1300–1306. https://doi.org/10.1093/bioinformatics/btp139

Krajewski, W. F., Ceynar, D., Demir, I., Goska, R., et al. (2017). Real-time flood forecasting and information system for the state of Iowa. *Bulletin of the American Meteorological Society*, *98*(3), 539–554. https://doi.org/10.1175/BAMS-D-15-00243.1

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall-runoff modelling using long-short-term-memory (LSTM) networks. Hydrology and Earth System Sciences Discussions, (May), 1–26. https://doi.org/10.17605/OSF.IO/QV5JZ

Krause, P., Boyle, D. P., & Bäse, F. (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*, *5*, 89–97. https://doi.org/10.5194/adgeo-5-89-2005

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., et al. (2016). Ask me anything: Dynamic memory networks for natural language processing. Retrieved from http://proceedings.mlr.press/v48/kumar16.pdf

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lin, Y. (2011). GCIP/EOP Surface: Precipitation NCEP/EMC 4KM Gridded Data (GRIB) Stage IV data. Version 1.0. UCAR/NCAR - Earth Observing Laboratory. https://doi.org/https://doi.org/10.5065/D6PG1QDD

Maier, H. R., Jain, A., Dandy, G. C., & Sudheer, K. P. (2010). Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environmental Modelling & Software*, *25*(8), 891–909. https://doi.org/10.1016/J.ENVSOFT.2010.02.003

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, *50*(3), 885–900. https://doi.org/10.13031/2013.23153

Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. *Water*, *10*(11), 1536. https://doi.org/10.3390/w10111536

Nagesh Kumar, D., Srinivasa Raju, K., & Sathish, T. (2004). River flow forecasting using recurrent neural networks. *Water Resources Management*, *18*(2), 143–161. https://doi.org/10.1023/B:WARM.0000024727.94701.12

Neitsch, S. L., Arnold, J. G., Kiniry, J. R., & Williams, J. R. (2011). Soil and Water Assessment Tool theoretical documentation version 2009. Retrieved from http://oaktrust.library.tamu.edu/handle/1969.1/128050

Nhita, F., Saepudin, D., Adiwijaya, & Wisesty, U. N. (2016). Comparative study of moving average on rainfall time series data for rainfall forecasting based on evolving neural network classifier. In Proceedings - 2015 3rd International Symposium on Computational and Business Intelligence, ISCBI 2015 (pp. 112–116). IEEE. https://doi.org/10.1109/ISCBI.2015.27

Olah, C. (2015). Understanding LSTM networks. Retrieved November 12, 2018, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Ömer Faruk, D. (2010). A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering Applications of Artificial Intelligence*, *23*(4), 586–594. https://doi.org/10.1016/J.ENGAPPAI.2009.09.015

Puckridge, J. T., Walker, K. F., & Costelloe, J. F. (2000). Hydrological persistence and the ecology of dryland rivers. *Regulated Rivers: Research & Management*, *16*(5), 385–402. https://doi.org/10.1002/1099-1646(200009/10)16:5<385::AID-RRR592>3.0.CO;2-W

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195–204. https://doi.org/10.1038/s41586-019-0912-1

Riad, S., Mania, J., Bouchaou, L., & Najjar, Y. (2004). Rainfall-runoff model usingan artificial neural network approach. *Mathematical and Computer Modelling*, *40*(7–8), 839–846. https://doi.org/10.1016/j.mcm.2004.10.012

Sajikumar, N., & Thandaveswara, B. S. (1999). A non-linear rainfall–runoff model using an artificial neural network. *Journal of Hydrology*, *216*(1–2), 32–55. https://doi.org/10.1016/S0022-1694(98)00273-X

Sermet, Y., & Demir, I. (2018). An intelligent system on knowledge generation and communication about flooding. *Environmental Modelling & Software*, *108*, 51–60.

Sit, M., & Demir, I. (2019). Decentralized flood forecasting using deep neural networks. Retrieved from http://arxiv.org/abs/1902.02308

Sit, M. A., Koylu, C., & Demir, I. (2019). Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and spatial analysis: a case study of Hurricane Irma. *International Journal of Digital Earth*, *12*(11), 1–25.

Sun, A. Y., Wang, D., & Xu, X. (2014). Monthly streamflow forecasting using Gaussian process regression. *Journal of Hydrology*, *511*, 72–81. https://doi.org/10.1016/J.JHYDROL.2014.01.023

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks, 3104–3112. Retrieved from https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks

U.S. Geological Survey. (2016). National Water Information System data available on the World Wide Web (USGS Water Data for the Nation). https://doi.org/https://doi.org/10.5066/F7P55KJN

Weber, L. J., Muste, M., Bradley, A. A., Amado, A. A., Demir, I., Drake, C. W., et al. (2018). The Iowa Watersheds project: Iowa's prototype for engaging communities and professionals in watershed hazard mitigation. *International Journal of River Basin Management*, *16*(3), 315–328. https://doi.org/10.1080/15715124.2017.1387127

10.1029/2019WR025326

Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D. Young, S. (2015). Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. Retrieved from http://arxiv.org/abs/1508.01745

Xiang, Z., Montas, H., Shirmohammadi, A., Leisnham, P., & Brubaker, K. (2018). Impact of climate change on critical source areas in a Chesapeake Bay watershed. *2018* Detroit, Michigan July 29 - August 1, 2018, 1. https://doi.org/10.13031/aim.201801831

Yan, J., Jin, J., Chen, F., Yu, G., Yin, H., & Wang, W. (2018). Urban flash flood forecast using support vector machine and numerical simulation. *Journal of Hydroinformatics*, *20*(1), 221–231. https://doi.org/10.2166/hydro.2017.175

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, *561*, 918–929. https://doi.org/10.1016/j.jhydrol.2018.04.065