



Research papers

Improved runoff forecasting performance through error predictions using a deep-learning approach

Heechan Han^a, Ryan R. Morrison^{b,*}^a Blackland Research and Extension Center, Texas A&M AgriLife, Temple, TX, USA^b Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, CO, USA

ARTICLE INFO

This manuscript was handled by Sally Elizabeth Thompson, Editor-in-Chief, with the assistance of Bellie Sivakumar, Associate Editor

Keywords:

Deep learning
Error prediction
LSTM-s2s
National Water Model

ABSTRACT

Accurate runoff prediction is critical for various fields of hydrology, agriculture, and environmental studies. Numerous hydrologic models have been developed and demonstrate good performances in runoff simulation. However, errors are inherent in forecasted runoff predictions, which can cause uncertainty in real-time flood warning systems. In order to improve the predictive performance of hydrologic modeling, this study used a deep learning approach as a post-processor to correct for errors associated with hydrologic data. The proposed model uses the long short-term memory model with sequence-to-sequence structure as a post-processor to improve runoff forecasting. Specifically, the deep learning approach was used to estimate errors in forecasted hourly runoff provided from National Water Model in Russian River basin, California, United States. Error prediction in hourly runoff with lead times between 1 and 18 h were developed using observed precipitation and errors from upstream stream gages to improve the predictive performance of National Water Model. The predictive performance of the model was evaluated using numerous statistical metrics, and results show that the long short-term memory model with sequence-to-sequence post-processor improved runoff predictions compared to standalone results from the National Water Model. Statistical values of percent bias decreased from a range of -60%–80% to -15%–10% when the post-processor model was used, and similarly root mean square errors of runoff prediction decreased from 120 cms to 20 cms. Thus, this study demonstrates the power of deep learning model to improve hydrologic modeling results, especially those with short forecasting lead times.

1. Introduction

Accurately simulating relationships between rainfall and runoff is important for various water management purposes, particularly flood forecasting (Shrestha and Solomatine, 2008; Neitsch et al., 2011; Hu et al., 2018; Fan et al., 2020; Xiang et al., 2020). Physical-based numerical hydrological models are often used to simulate nonlinear rainfall-runoff relationships, and these models are subsequently applied in various hydrologic applications, including short-term simulations for flood prediction and long-term simulations for drought analysis and water resources management (Ott et al., 1991; Lee et al., 2005; Wu et al., 2011; Kang and Sridhar, 2017; Wang et al., 2017). These models generally contain multiple parameters used to calibrate hydrological processes and to minimize output residuals or errors.

In hydrological models, errors inherent in the flow prediction results can lead to inaccurate hydrological analyses. Decision makers operating water-related infrastructures for flood and drought forecasting,

irrigation control, and other water management purposes require reliable hydrologic modeling results that consider uncertainties inherent in the results (Shrestha and Solomatine, 2008). Proper error estimation and prediction can enhance the reliability and credibility of hydrologic modeling outputs for water management and improve our understanding of error propagations within modeling frameworks (Krzysztofowicz, 2001).

Differences between modeled results and observed data (e.g. differences between modeled streamflow estimates and measured discharge) can be caused by four general sources within a hydrologic model: (a) uncertainties or inaccuracies in modeling input data; (b) uncertainties or inaccuracies in modeled outputs used for calibration; (c) uncertainties or inaccuracies in modeling parameters; and (d) uncertainties caused by imperfect model structures (Refsgaard and Storm, 1990; Shrestha and Solomatine, 2008). A major contributor to errors caused by any of these types of uncertainties is the difficulty in accurately measuring the spatial and temporal variability in hydrologic modeling inputs across large

* Corresponding author.

E-mail address: Ryan.Morrison@colostate.edu (R.R. Morrison).

modeling domains (e.g. large watershed or groundwater aquifers) (Shrestha and Solomatine, 2008).

Numerous methods have been used to analyze uncertainties in hydrological models. These methods include analytical and approximation methods (Rosenblueth, 1975; Harr, 1989; Melching, 1992; Tung, 2011; Maskey et al., 2004; Montanari, 2007), Monte Carlo-based methods (Kuczera and Parent, 1998; Hong et al., 2006; Vrugt et al., 2008), Bayesian methods and generalized likelihood uncertainty estimation (GLUE) methods (Krzysztofowicz, 1999; Jin et al., 2010; Li et al., 2010), and Fuzzy theory-based methods (Maskey and Guinot, 2003; Huang et al., 2010). However, the majority of these methods deal only with a single source of uncertainty, and most methods assume that the model structure is accurate, and that input data is free from uncertainty. These limitations make it difficult to consider the inherent uncertainties of input data and numerical approaches for limiting the effects of input errors on model outputs. According to Abebe and Price (2003), uncertainties associated with the quality of "upstream" input data affect the calibration of the model parameters, and input errors translate through the model to its outcome. As long as there are unaccounted uncertainties in input data, it is difficult to expect accurate rainfall-runoff simulations through hydrological models.

For instance, in a short-term rainfall-runoff model, which may predict the amount and pattern of runoff for a flood warning system, accurate runoff forecasting is essential. Since the flood warning level is determined according to the forecasted results from the hydrological model, it is crucial to maintain high model accuracy. Moreover, it is necessary to manage the errors in real time because the short-term hydrological models provide runoff predictions every hour. Ideally, it is possible to minimize the uncertainty of the predicted runoff by predicting the errors that can be generated from various sources in advance.

Many studies have examined how uncertainty from upstream sources affect the errors in model outcomes (Kobold and Sušelj, 2005; Haydon and Deletic, 2009; Arnaud et al., 2011; McMillan et al., 2011). For example, Muñoz et al. (2014) analyzed how the uncertainty in precipitation affects the runoff errors simulated by a hydrological model by using Monte Carlo simulations, and they found that the uncertainty in the precipitation estimation processes had a significant effect on the modeling output errors. Obviously, uncertainty in the precipitation data during a rainy period had a larger impact on output errors compared to periods with less precipitation. Another study, led by Datta and Bolisetti (2016), used the precipitation multiplier method to study the impacts of precipitation uncertainty on model error, and the study concluded that differences between estimated precipitation and true precipitation have significant effects on the errors in model prediction. Specifically, the errors in model prediction resulted in underestimated discharges for high-flow events and overestimated discharges for low-flow events. Thus, it is essential to consider the impact of uncertainty in upstream data sources when minimizing the output errors of hydrologic models.

Recently, deep learning models have been applied for various purposes in the field of hydrology. Deep learning models use large datasets to analyze and predict non-linear relationships between data components, such as rainfall-runoff relationships. Such models have significantly contributed to the advancement of hydrological analyses since they provide high-quality and cost-effect modeling results (Mosavi et al., 2018). In addition, the deep learning models have shown predictive powers with fewer parameters compared to the physical-based models (Solomatine and Ostfeld, 2008; Castelletti et al., 2010).

Deep learning models have been widely used in various forecasting applications, including predictions of precipitation (Lin et al., 2013; Agrawal et al., 2019; Sønderby et al., 2020), runoff (Yilmaz and Muttgil, 2014; Hu et al., 2018; Fan et al., 2020; Xiang et al., 2020), soil properties (Feng et al., 2019), groundwater levels (Sahoo et al., 2017), and river stages (Choi et al., 2020). In addition, deep learning models can be used in combination with two or more models (Kim et al., 2019) and can be used to improve the performance of physical-based models as they are used complementarily (Abebe and Price, 2003).

Not only are deep learning models effective in predicting hydrologic variables, but they can also be used to analyze and predict how errors in model outputs are affected by various sources of uncertainty. Several studies have analyzed the uncertainties of hydrological models using deep learning model to improve the predictive performance of the models. Abebe and Price (2003) presented a complementary framework to manage uncertainty in hydrological models using an artificial neural network (ANN) model. Their study focused on the relationship between input data (e.g. precipitation) and error in output (e.g. runoff) of the model, and they concluded that the usage of ANN model for error predictions in hydrological models can significantly improve model performance. Moreover, Wu et al. (2018) successfully applied a random forest model to predict the errors based on relationships between input variables (e.g. precipitation, temperature) and output (e.g. runoff) error, and, similarly, Shrestha and Solomatine (2009) applied various machine learning models to estimate the prediction intervals of outputs and concluded that machine learning approaches are effective methods to estimate model uncertainties. In addition, deep learning models have recently been used as a post-processor for minimizing runoff errors and improving the performance of hydrological models (Frame et al., 2020; Nearing et al., 2020).

Previous studies only considered error sources related to uncertainties within the model, such as choice of input variables, parameters, and model structure, so it is difficult to account for the influences of upstream data uncertainty on errors in the modeling output. By building upon previous data-driven modeling studies, this study developed a post-processor framework using deep learning approach and evaluated the framework to reduce hydrologic modeling errors by explicitly considering uncertainties in upstream runoff data. The Long-Short Term Memory approach, which is a deep learning model, was combined with a sequence-to-sequence learning structure (hereafter referred to as LSTM-s2s) to predict the errors in the forecasted runoff from a hydrological model. Recently, Xiang et al. (2020) evaluated LSTM-s2s model to predict the runoff and suggested the LSTM-s2s model for hydrological predictions. Also, Frame et al. (2020) applied Long-Short Term Memory approach (LSTM) to improve the performance of National Water Model (NWM) for runoff simulation by considering NWMs' output as input variables of LSTM model. However, these studies focused on only runoff simulation by using LSTM-based deep learning approach, instead of the uncertainty of modeling results.

The novelty of this study is to hybrid use of deep learning approach and physical-based model for predicting the model errors in outcomes, instead of the runoff itself. The specific objectives of this were to: (i) analyze errors in forecasted runoff from the physical-based hydrologic model, (ii) develop an error correction framework using a deep learning model as a post-processor framework, and (iii) investigate how the developed post-processing framework improves the predictive performance of hydrological model.

2. Materials and methods

2.1. Study area

This study used the Russian River basin, located in California, USA, as a study area. The Russian River basin is located on the west coast of the United States (Fig. 1) and receives an average annual precipitation of 925 mm with more than 80% of the annual precipitation occurring between November and March. Intense precipitation is often generated in this region from extratropical cyclones or jet streams from the Pacific Ocean, which have historically caused flood damages during the wet season (Ralph et al., 2006; Johnson et al., 2016; Han et al., 2019).

The basin has a drainage area of approximately 3,850 km² with elevations ranging from 50 m to nearly 800 m. Two reservoirs have been constructed in the basin, Mendocino and Sonoma Reservoirs, which are regulated by the Coyote Valley and Warm Springs dams for irrigation and flood control. Because the Russian River is one of the most flood-

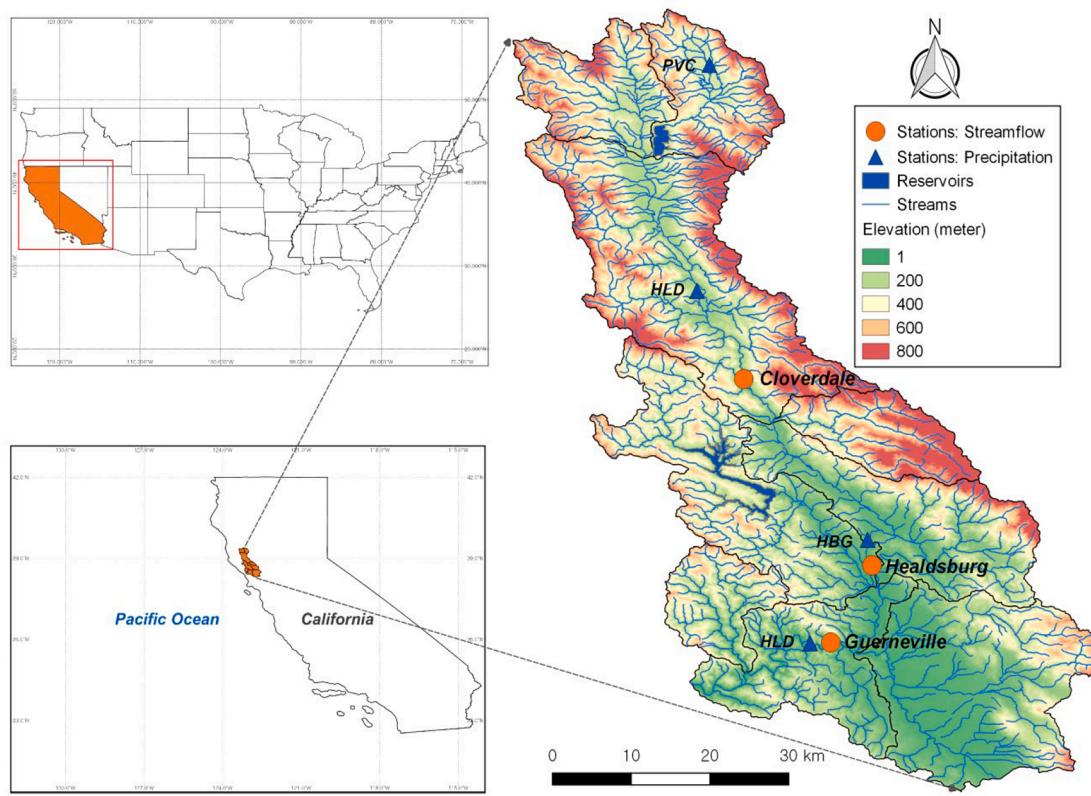


Fig. 1. Location of the Russian River basin and hydrologic observation stations located within the basin.

prone rivers in California due to its unique combination of geography and climatologically heavy precipitation, the basin is ideally suited to evaluate and reduce errors associated with runoff modeling and flood forecasting. Fig. 1 shows the precipitation stations, three United States Geological Survey (USGS) streamflow gages, and the stream channel network within the basin.

2.2. National water model

The NWM was developed to simulate and forecast hydrological processes, including streamflow runoff, throughout the Continental United States (CONUS). The NWM uses observed datasets from more than 8,000 USGS gages to forecast various hydrological variables, including soil moisture, surface runoff, snow water equivalent, and other variables for 2.7 million locations in the CONUS. The core system of the NWM contains the National Center for Atmospheric Research (NCAR)-supported community Weather Research and Forecasting Hydrologic model (WRF-Hydro) (<https://water.noaa.gov/about/nwm>). The model provides forecasted outputs for three geospatial features, including river channels, land types, and reservoirs. The types of channels and reservoirs are based on the U.S. NHDPlus dataset, and the land types are based on a 1 km- and 250 m-grid system covering the CONUS. For example, the NWM provides land surface variables, such as evapotranspiration, snow water equivalent, and snow depth at a 1 km-grid resolution and ponded water depth and soil saturation outputs at a 250 m-grid resolution (Han et al., 2019). The NWM has three forecasting configurations, including short-range, medium-range and long-range forecasting. Table 1 shows the features of each.

In this study, hourly streamflow forecasted from short-range configuration from 2019 to 2020 were used to estimate errors from the NWM and compare them with the results from the deep learning model (i.e., LSTM-s2s). The short-range configuration produces hourly deterministic forecasts of streamflow and hydrologic states with a lead

Table 1
Features of NWM configurations.

Configurations	Frequency	Forecast duration	Forecast step
Short-range	Hourly	0–18 h	1 h
Medium-range	Daily	0–10 days	3 h
Long-range	Daily	0–30 days	6 h

(https://water.noaa.gov/about/output_file_contents).

time of 1–18 h in 1-hour increments.

2.3. Data descriptions

Datasets for this study include forecasted runoff data with lead times of 1 to 18 h in 1-hour increments from the NWM and observed runoff at three USGS stations located in the Russian River basin: USGS 11,463,000 at the Russian River basin near Cloverdale (Cloverdale station), USGS 11,464,000 at the Russian River near Healdsburg (Healdsburg station), and USGS 11,467,000 at the Russian River near Guerneville (Guerneville station). The Guerneville station is located at the outlet point of the watershed. Additionally, hourly precipitation data from NOAA Physical Sciences Laboratory (<https://www.psl.noaa.gov/data/>) at the Potter Valley Central (PVC), Hopland (HLD), Rio Nido (ROD), and Healdsburg (HBG) stations were used as input data to the LSTM-s2s model. Datasets from 2019 were used for training the LSTM-s2s model, and datasets from 2020 were used to test of the model performance. Table 2 describes the various datasets used in this study.

2.4. Long-Short term memory with sequence to sequence learning

The LSTM model, introduced by Hochreiter and Schmidhuber (1997), is a recurrent neural network (RNN) deep learning model. The LSTM model is an effective method for solving gradient-vanishing

Table 2
Features of datasets used in this study.

Datasets (Source)	Stations (USGS)	Lat	Lon	Time-step
Streamflow (waterdata.usgs.gov)	Cloverdale (11463000)	38.8794	-123.0525	Hourly
	Healdsburg (11464000)	38.6133	-122.8352	Hourly
	Guerneville (11467000)	38.5086	-122.9266	Hourly
Precipitation (https://psl.noaa.gov/data/)	PVC	39.3209	-123.1027	Hourly
	HLD	39.0030	-123.1209	Hourly
	ROD	38.5073	-122.9565	Hourly
	HBG	38.6529	-122.8732	Hourly

problems generated in RNN models and for handling long-term time-dependent data. Since the training process of LSTM model is based on numerous parameters, the training process may be longer than other data-drive models, but it has the advantage of providing results with high accuracy (Hu et al., 2018). The LSTM model has been applied for linguistic translation, speech recognition, and time-series predictions, and recently also to the prediction of several hydrological variables (Soltan et al., 2016; Hu et al., 2018; Kratzert et al., 2018; Fan et al., 2020; Xiang et al., 2020). Fig. 2 shows the conceptual diagram of LSTM model.

The LSTM model consists of many hidden layers called memory cells, and each memory cell maintains the state of information at time t (Fig. 2). Each memory cell also includes three nonlinear gates—the forget gate (f_t), input gate (i_t), and output gate (O_t)—that control the

flow of information between cells. The LSTM operates the algorithms from an input sequence data x_t to the final outcome O by looping through equations (1) – (6) with parameter values of $C_0 = 0$ and $h_0 = 0$ indicating initial cell state and hidden state generated from first cell (Adeyemi et al., 2018). The equations used to control information flow through the LSTM gages are:

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

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

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (5)$$

$$h_t = O_t \times \tanh(C_t) \quad (6)$$

where, σ is the non-linear activation function. W_f , W_i , W_o and W_C are weight values for the forget gates, input gates, output gates and memory cells, h_{t-1} denotes output data from a previous cell, x_t is input data for the current time step, and b_f , b_i and b_o are bias vectors of each gate, respectively. In addition, C_t is the state of any cell generated from the activation function. In this study, ReLU was used as the activation function.

The purpose of the forget gate (equation 1) is to determine how much information from the previous block is maintained, and the gate determines this value by applying output from the previous cells and

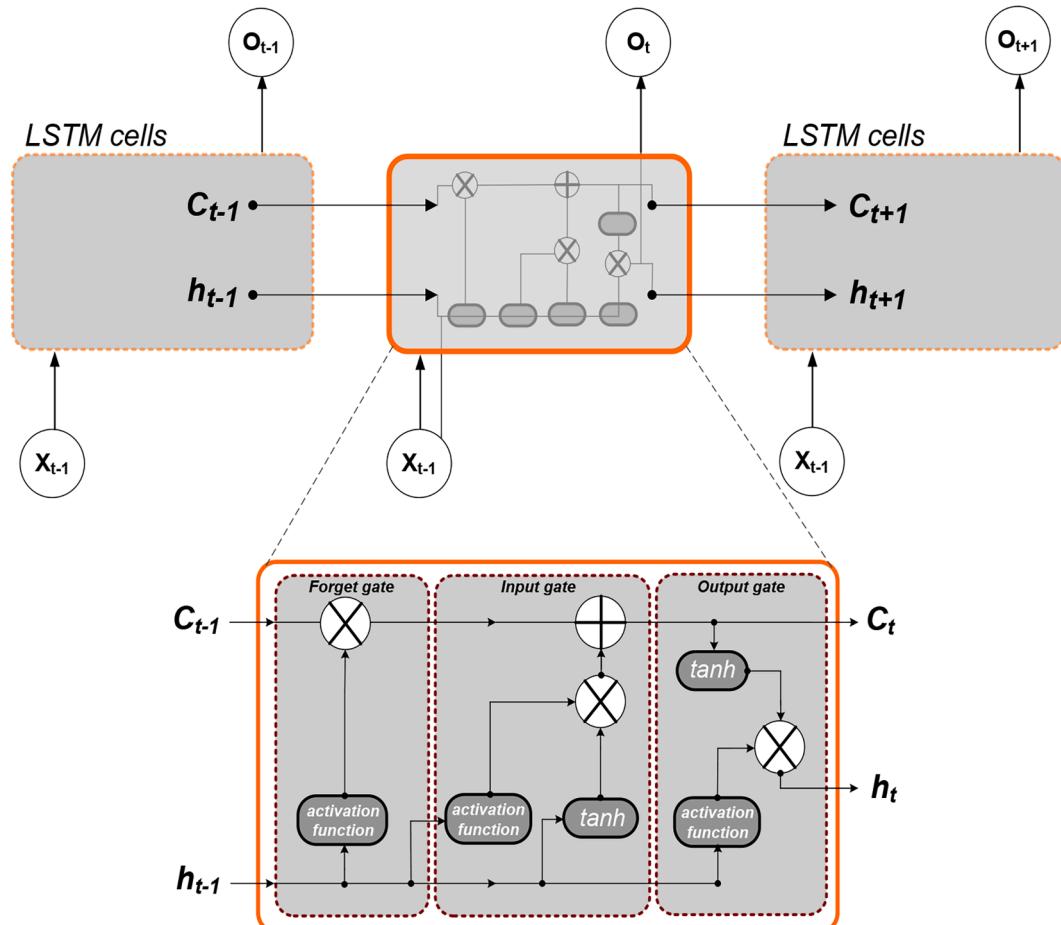


Fig. 2. Conceptual diagram of LSTM model.

current input data to the activation function. The input gate (equation 2) is able to determine which of the new information to store in a cell, and the output gate (equation 3) is used to determine the final output value from the information stored in the cell.

However, a limitation of LSTM models is that equal time steps for the model inputs and outcomes are required (Xiang et al., 2020). Combining the LSTM with a sequence-to-sequence learning model (LSMT-s2s), or Encoder-Decoder model, was developed to solve this limitation using a structure based on different time steps of input and output variables (Cho et al., 2014).

The main difference between the both of LSTM and LSTM-s2s is the number of time steps considered for the input and output variables. Therefore, there is an explicit benefit that data based on different time sequence can be analyzed more efficiently through an LSTM-s2s model. Generally, LSTM-s2s models are applied in such fields as text translation, speech recognition, and image analysis, which require different time sequence (Xiang et al., 2020). Similarly, LSTM-s2s models can be used in hydrological time series predictions, such as short-term runoff and flash flood forecasting (Kao et al., 2020; Xiang et al., 2020). Fig. 3 shows the conceptual diagram of a LSTM-s2s model.

The LSTM-s2s model process is similar to the traditional LSTM model, but it consists of three structures, known as the encoder, encoded vector and decoder stages. An encoder stage uses the information from given datasets as input variables, and output from the encoder stage with m time steps can be stored in a encoded vector. The encoded vector stage is the final hidden state given from the encoder stage. It aims to help the decoder predict output values accurately by condensing the information produced from the encoder part. Also, it can be the initial hidden state of the decoder stage of the model. In the decoder stage, it is then used to translate data stored in the encoded vector into predictions of output variables with n time steps. One of the beneficial features of LSTM-s2s models is that the time steps of encoder and decoder LSTM can be different. In this study, $X_1 \dots X_m$ represents the historical errors in runoff from the physical-based hydrological model and observed precipitation at upstream stations, and $O_1 \dots O_n$ denotes the predicted errors.

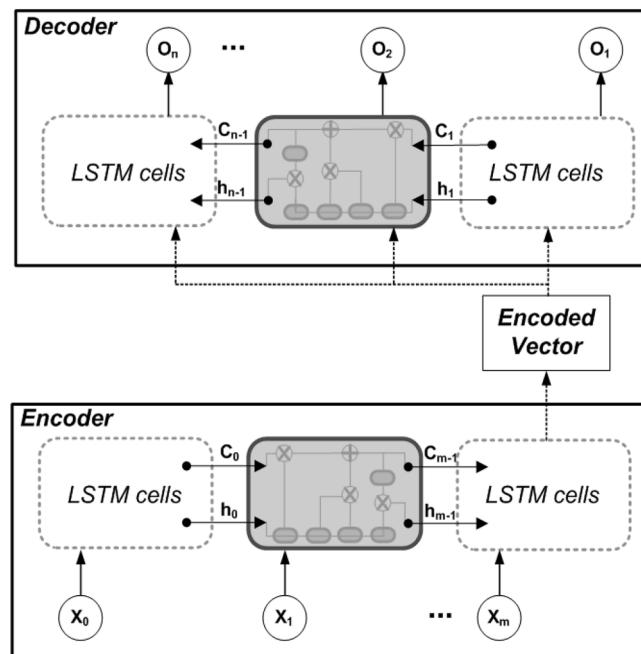


Fig. 3. Conceptual diagram of LSTM-s2s model. Details of each LSTM cell (grey box) in represented in Fig. 2.

2.5. Model design

In this study, a LSTM-s2s model was used to improve runoff forecasting with lead times between 1 and 18 h using runoff errors from upstream and observed precipitation as input data. Fig. 4 illustrates the model design, which includes the following five components: 1) collecting forecasted and observed runoff time series data; 2) calculating errors between forecasted and observed datasets; 3) training the LSTM-s2s model using the calculated errors and observed data; 4) predicting future model errors using the LSTM-s2s model; 5) improving the forecasted runoff and evaluating the forecast performance. Errors were estimated using equation (7) for each time step between 1 and 18 h.

$$\text{error}(\%) = \frac{(\text{Runoff}_{\text{obs}} - \text{Runoff}_{\text{sim}})}{\text{Runoff}_{\text{obs}}} \times 100 \quad (8)$$

where, $\text{Runoff}_{\text{obs}}$ and $\text{Runoff}_{\text{sim}}$ denote the observed and forecasted runoff from the NWM. Estimated errors in each time step are used as input variables of the LSTM-s2s model for training and error predictions. The overall model can be formulated as follows:

$$\text{error}_{t+m} = f(\text{error}_{\text{upstream}, t-1, t-2, \dots, t-n}, \text{precipitation}_{t-1, t-2, \dots, t-n}) \quad (8)$$

In this model algorithm, time series input, including errors in forecasted runoff at three stations (two upstream stations and the outlet point) and observed precipitation at four ground-based gauges, were used to predict the error for each time steps.

Four metrics were used for evaluating the performance of the model design: coefficient of correlation (CC), root mean square error (RMSE), percent bias (PBIAS), and Nash-Sutcliffe Efficiency (NSE).

CC is calculated as:

$$\text{CC} = \frac{\sum(y_e - \bar{y}_e)(y_o - \bar{y}_o)}{\sqrt{\sum(y_e - \bar{y}_e)^2} \sqrt{\sum(y_o - \bar{y}_o)^2}} \quad (9)$$

where, y_e and y_o indicate the simulated and observed runoff, \bar{y}_e and \bar{y}_o denote the average simulated and observed runoff. Values of CC range from -1 to 1 and describe a measure of how well the outputs are simulated by the model. A CC value of 0 indicates that there is no correlation between observed and simulated runoff.

RMSE and PBIAS are calculated as:

$$\text{RMSE} = \sqrt{\frac{\sum(y_o - y_e)^2}{m}} \quad (10)$$

$$\text{PBIAS} = \frac{\sum y_o - \sum y_e}{\sum y_o} \times 100(%) \quad (11)$$

Values of RMSE ranges from 0 to $+\infty$ and describes how well the simulated value matches to the observed value. The zero value of RMSE means the modelled and observed values are the same. The PBIAS denotes the ratio of difference between sums of simulated and observed values to the sum of observed values.

NSE is calculated as:

$$\text{NSE} = 1 - \frac{\sum(y_e - y_o)^2}{\sum(y_o - \bar{y}_o)^2} \quad (12)$$

Values of NSE represents the predictive power of the model and range from $-\infty$ to 1 . NSE values near 1 indicate excellent model performance.

3. Results

3.1. National water model performance for runoff forecasting

It was necessary to first evaluate the runoff predictions from the NWM before testing prediction improvements using the LSTM-s2s model. A total of 11,647 values of predicted runoff from the NWM

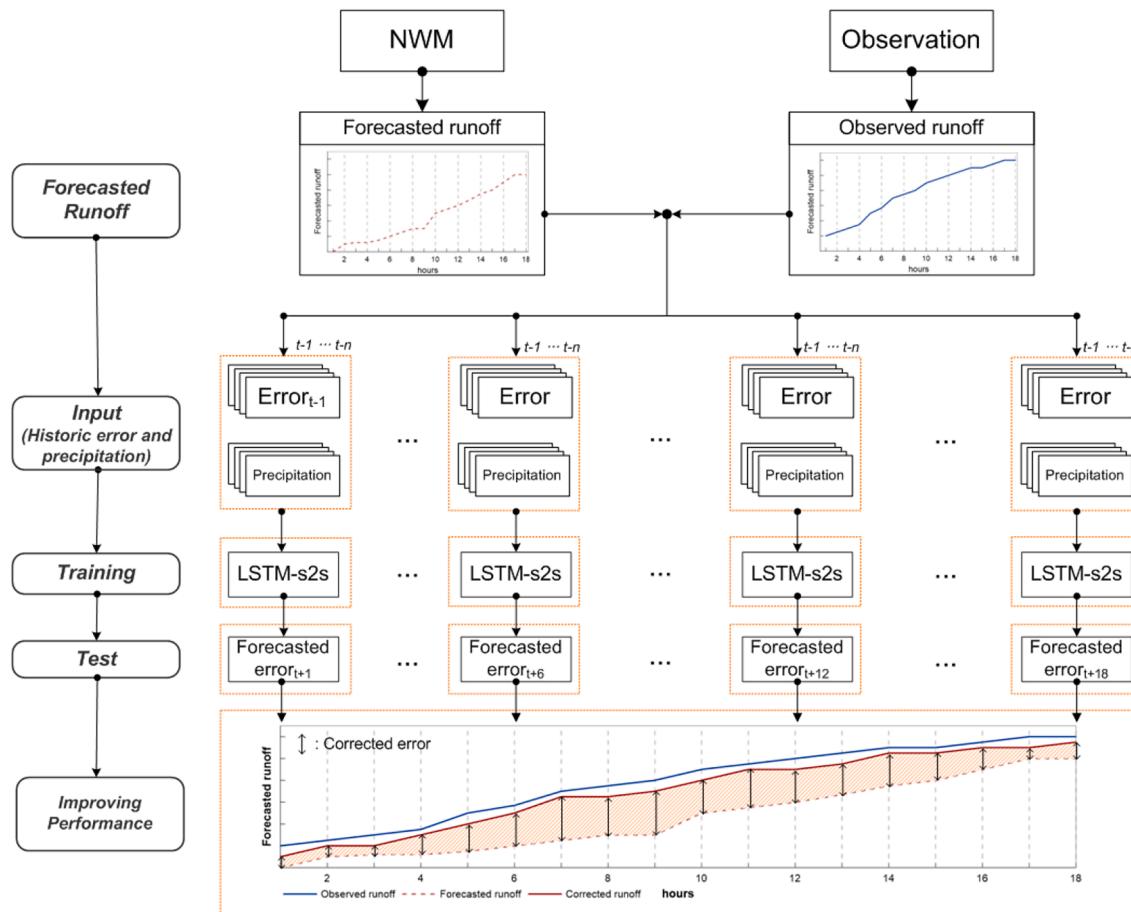


Fig. 4. Flow chart of this study which is representing five steps from estimating errors in forecasted runoff to the improvement of forecasting performance.

were compared with observed runoff data collected at the USGS stations. Evaluation metrics of CC, NSE, PBIAS, and RSME were used to compare runoff predicted by the NWM and observed runoff at each of the three USGS gages, and the cumulative distributions of each metric were plotted to assess the NWM predictive performance (Fig. 5). As shown in the Fig. 5, it was found that the performance of the NWM according to the evaluation metrics was similar at each USGS station, though runoff predictions at the Healdsburg station were the most accurate based on NSE and PBIAS distributions. Overall, according to PBIAS values, the NWM runoff predictions at each USGS station tended to overestimate actual runoff, indicating that improvements to runoff predictions are necessary.

The forecasting performance of the NWM for each 1-hour time step between 1 and 18 h was also assessed using the evaluation metrics. As shown in Fig. 6, the predictive performance of the NWM at the Guerneville station tends to decrease with growing lead times until the performance stabilizes for lead times greater than approximately 12 h (CC ≈ 0.66 , NSE ≈ 0.55 , PBIAS $\approx 10\%$, and RMSE ≈ 40 cms). These results show the necessity of improving the predictive performance of NWM for effective flood forecasting, especially at longer lead times.

3.2. Error prediction results

3.2.1. Model training

The LSTM-s2s model was trained by evaluating its performance after adjusting the parameters for input data lag times, number of LSTM neuron cells in each layer, and batch sizes in each layer (Fan et al., 2020; Xiang et al., 2020). During model training, lag times, number of neurons, and batch sizes were adjusted to achieve optimal model

performance. Previous studies have also focused on adjusting these parameters during the LSTM model training period (Fan et al., 2020; Xiang et al., 2020).

In our model, observed hourly precipitation at four stations (PVC, HLD, ROD, and HBG) and residuals in runoff at two upstream USGS stations (Cloverdale and Healdsburg) were used as input data. To evaluate the sensitivity of the model to variations in lag time of input data, lag times between 6 and 24 h were tested from the upstream precipitation and USGS stations, and model performance was evaluated at the most downstream USGS station (Guerneville). As shown in Fig. 7(a), the results indicate that the model performs best with lag times of 6 h (NSE = 0.95, CC = 0.97, PBIAS = 47%), and models with lags between 6 and 12 h perform similarly based on CC and NSE values but have higher PBIAS values as lag times increase. Models with lag times between 12 and 18 h have relatively low performance, likely because the time to concentration between the upstream stations and basin outlet varies between 10 and 18 h according to time differences in time to runoff peaks between each station. This indicates that other hydrologic processes, including soil moisture and evapotranspiration, are also important for predicting runoff with longer time-steps (Xiang et al., 2020), though not considered in this study.

The sensitivity of model performance to the number of neurons and batch sizes is shown in Fig. 7 (b) and (c). These two parameters were adjusted from 32 to 256 and tested at the Guerneville station. The model with 32 number neurons and a batch size of 128 performs best (CC = 0.96 (0.75), NSE = 0.91 (0.71), PBIAS = 59.4% (63.9%), RMSE = 1.4 cms (2.4 cms) for 32 number neurons (128 batch size)). The model with a batch size of 32 also showed acceptable performance with high values of CC and NSE, but it has larger PBIAS values compared to other models

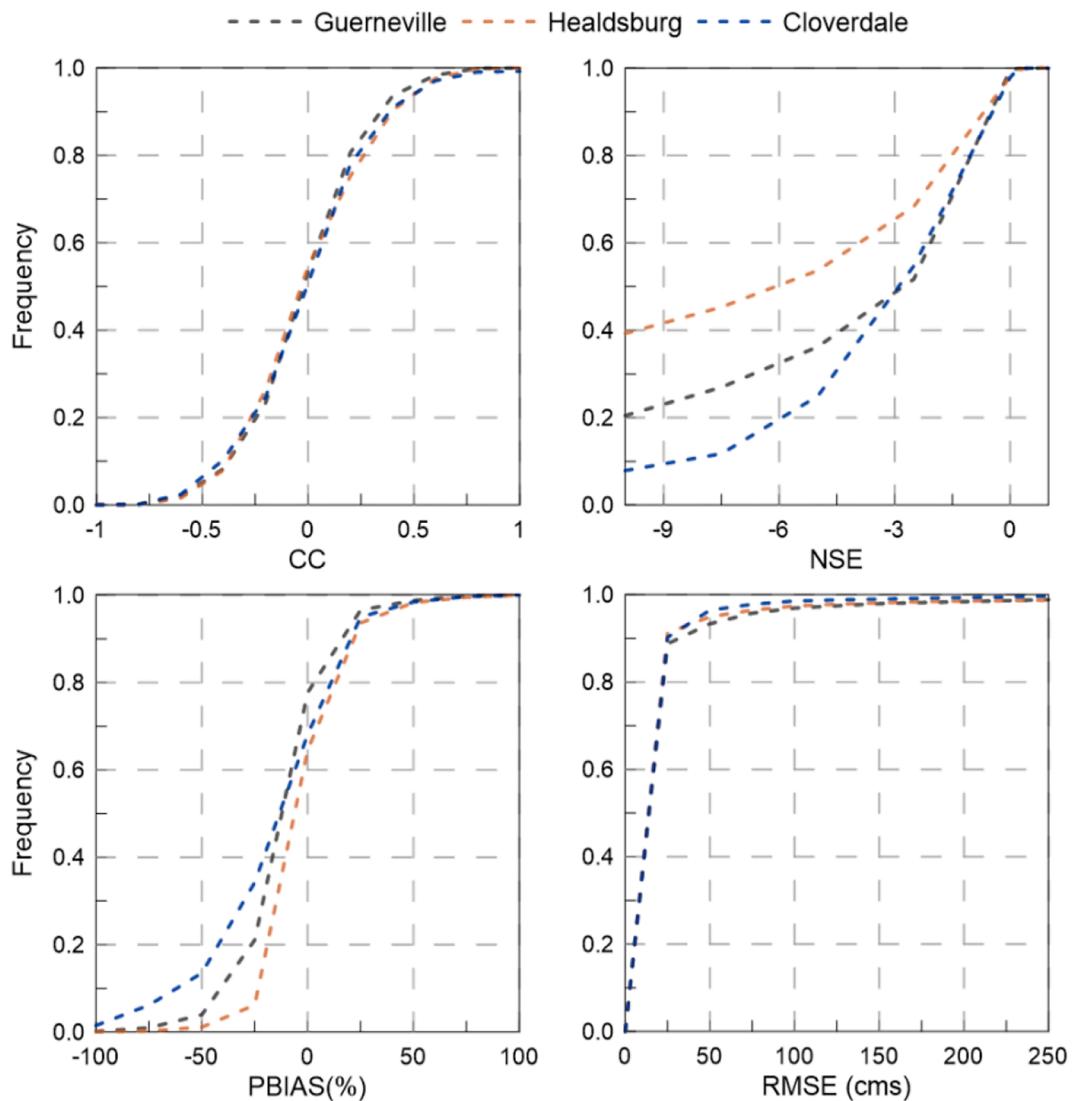


Fig. 5. Cumulative distributions of model performance estimated as CC, NSE, PBIAS, and RMSE for three stations (Cloverdale, Healdsburg, and Guerneville).

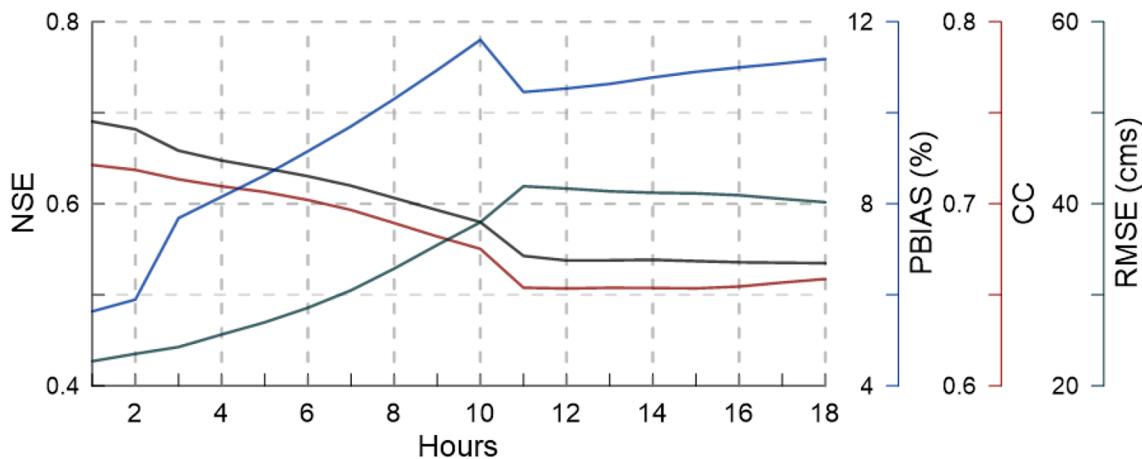


Fig. 6. Evaluation results of predictive performance using four error indices for 1 to 18 h of lead time at Guerneville station.

with different batch sizes. Based on these model training results, this study used a lag time of 6 h, number of neurons of 32, and batch size of 128 as hyperparameters of the LSTM-s2s model.

3.2.2. Error prediction using LSTM-s2s model

The trained LSTM-s2s model was used to predict runoff errors at the Guerneville station with forecasting lead times of 1 to 18 h in 1-hour increments. Results for each lead time (Fig. 8) indicate a strong model

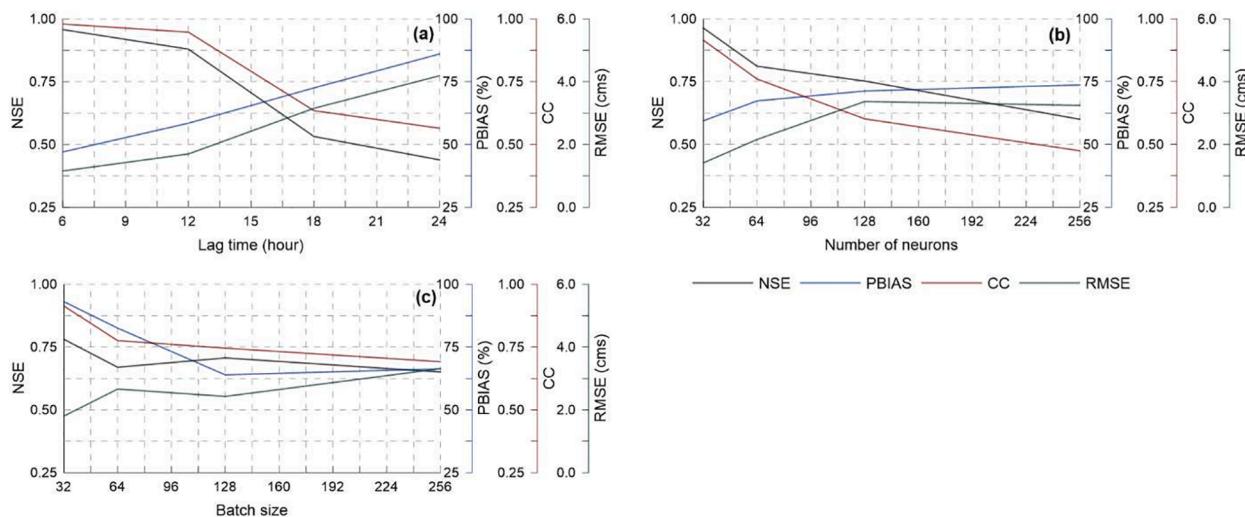


Fig. 7. Model training results based on three parameters including (a) lag time, (b) number of neurons, and (c) batch size using four evaluation metrics (NSE; black line, PBIAS; blue line, CC; red line, and RMSE; green line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

performance with average $R^2 = 0.97$. Distributions of error results range from -100% to 100% for 12–18 h lead times and -50% to 100% for 1–6 h lead times, and a high density (red points in Fig. 8) of error predictions between -20% to 0% . In addition, as a lead time increased, the distributions of errors tended to increase.

The statistical results for the overall predictive performance of the LSTM-s2s model for lead times between 1 and 18 h are listed in Table 3. As shown in Table 3, the LSTM-s2s model provided high performance for error prediction with average CC of 0.95, NSE of 0.86, RMSE of 5.31, and PBIAS of 2.11%. It was found that the PBIAS values were positive, indicating predicted errors were overestimated compared to the observations. Also, NSE values were 0.72 or greater, indicating acceptable model results (Xiang et al., 2020; Moriasi et al., 2007). Thus, from the statistical results, the LSTM-s2s model showed acceptable error prediction compared to the observed error, confirming its applicability for improving model performance as a post-processor.

3.3. Improved runoff forecasting performance

Predicted errors for each lead time from the LSTM-s2s model, described in section 3.2, were used to improve the accuracy of forecasted runoff measured at the Guerneville station. It was found that the LSTM-s2s model-based post-processor improved predictive performance of the NWM model. Fig. 9 shows results comparing the performance of the NWM with and without incorporating the error predictions from the LSTM-s2s model. Distributions of observed (grey markers) and corrected runoff (blue markers) were predominantly between 0 and 50 cms. The agreement between observed and predicted runoff from the NWM without LSTM-s2s error predictions ranged between $R^2 = 0.75$ to 0.95, whereas the agreement with LSTM-s2s error predictions ranged between $R^2 = 0.98$ to 0.99. Improvements in runoff predictions were shown in all lead times between 1 and 18 h (Fig. 9).

Results also show that the LSTM-s2s based post-processor improved the predictive performance of the NWM according to the metrics of CC, PBIAS, and RMSE. As shown in Fig. 10, evaluation metrics for runoff predictions of the NWM with (without) the LSTM-s2s were CC = 0–1 (-0.5–0.5), PBIAS = -15%–10% (-60%–80%), and RMSE = 0 cms–20 cms (0 cms–120 cms).

To highlight the effects of using the LSTM-s2s model to improve runoff forecasting, Fig. 11 shows two observed runoff hydrographs and the forecasting results of the NWM with and without the addition of the LSTM-s2s model for lead times between 1 and 18 h. Deviations between the runoff predictions from NWM and observations were large for all lead times, whereas the NWM with the LSTM-s2s model produced results that followed the same patterns of observed runoff. In addition, it was found that the fluctuation and deviation of the improved runoff were significantly lower during low-runoff periods than periods of high-runoff. As shown in Fig. 11 (a) and (b), the error in the predicted runoff compared to the observations was also larger in the NWM compared to the NWM with the LSTM-s2s model. For example, in Fig. 11 (a), the range of total errors (for 1–18 h) in runoff predicted from the NWM model ranged from -25% to 50% , whereas the range of errors in predicted runoff from the NWM with the LSTM-s2s model ranged from -3% to 3% for period of 2020-01-16 to 2020-01-20.

4. Discussions

This study demonstrates improvements in hydrologic runoff forecasting when a deep learning model is used to predict errors in runoff data at various lead times. Specifically, in the Russian River basin, it was shown that a LSTM-s2s post-processor model has the potential to improve the predictive performance of the NWM at a Guerneville USGS station, located near the outlet of the basin. Although this study used a relatively small basin as a case study to demonstrate the application of the LSTM-s2s model, it is expected that the deep learning model can be applied in other basins to improve the forecasting performance of the NWM. For instance, Frame et al. (2020) showed that LSTM based post-processor (without the s2s decoder) can be applied to improve the NWM simulation performance for daily runoff at 531 basins across the continental United States. They focused on predictions of daily runoff using NWM output data as input for a LSTM model. In addition, they directly predicted the runoff, which is different from the method of our study, which predicts the errors in forecasted runoff in advance. However, both studies have a same purpose of improving NWM runoff prediction performance using an LSTM-based post-processor. Both studies show that an LSTM post-processor can be used to significantly improve the

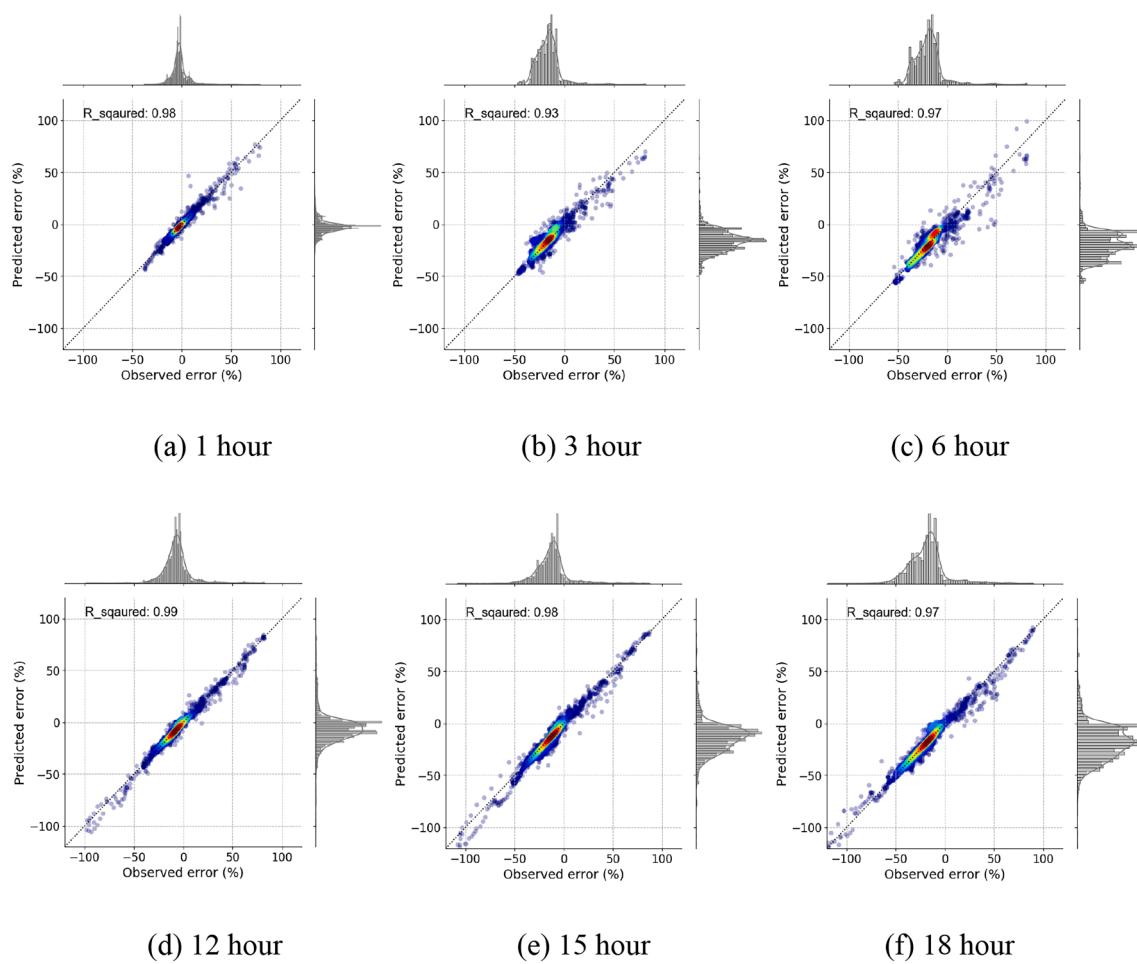


Fig. 8. Comparison results of predicted errors with actual values at various predictive lead times.

predictive ability of the NWM for daily and hourly runoff forecasting. However, the major difference between previous work and this study is that we present a method to improve the performance of hydrologic models, such as the NWM, by predicting and adjusting for errors in runoff through a post-processor, instead of directly predicting runoff, as performed in other previous studies (e.g., [Xiang et al. \(2020\)](#) and [Frame et al. \(2020\)](#)).

In this study, only observed precipitation and errors generated from upstream stations were considered as input variables, but further work should consider other variables that influence hydrologic routing, such as diversion infrastructure, reservoirs, and other management infrastructure, since these variables can impact modeling errors in NWM. For example, [Frame et al. \(2020\)](#) used other NWM variables, such as air temperature, radiation, vapor pressure, potential evapotranspiration,

and snow fraction, as inputs of their deep learning model to improve the representation of streamflow patterns. They showed the deep learning model can improve prediction accuracy of NWM. Although they used output states of the NWM as input data for their LSTM model, further work can consider actual errors in input variables and their effects on forecasted runoff.

The proposed post-processing model contains only precipitation and errors in runoff as input data. To apply the LSTM-s2s model to other regions with various meteorological characteristics, a new input selection process may be required to consider other hydrologic variables, such as snow accumulation, which is not observed for the Russian River basin. Thus, it is expected that several variables, including snow accumulation data, soil moisture data, and evapotranspiration data, can be considered to improve the predictive accuracy of the NWM model in

Table 3

The statistical results for the overall predictive performance of LSTM-s2s for error prediction for lead time between 1 and 18 h.

Lead time	CC	NSE	PBIAS (%)	RMSE	Lead time	CC	NSE	PBIAS (%)	RMSE
1	0.98	0.96	-11.83	2.05	10	0.89	0.72	-13.53	8.65
2	0.96	0.92	-30.75	3.03	11	0.99	0.98	-3.94	2.17
3	0.93	0.84	9.40	5.07	12	0.99	0.97	1.20	2.87
4	0.94	0.81	12.18	5.79	13	0.98	0.97	-4.63	3.24
5	0.85	0.53	30.71	9.59	14	0.98	0.95	-19.09	4.06
6	0.94	0.89	1.03	4.94	15	0.98	0.96	-4.94	4.06
7	0.87	0.68	16.82	8.51	16	0.95	0.84	32.91	7.92
8	0.94	0.87	-2.33	5.59	17	0.96	0.90	9.25	6.56
9	0.93	0.82	13.03	6.71	18	0.97	0.95	2.55	4.79

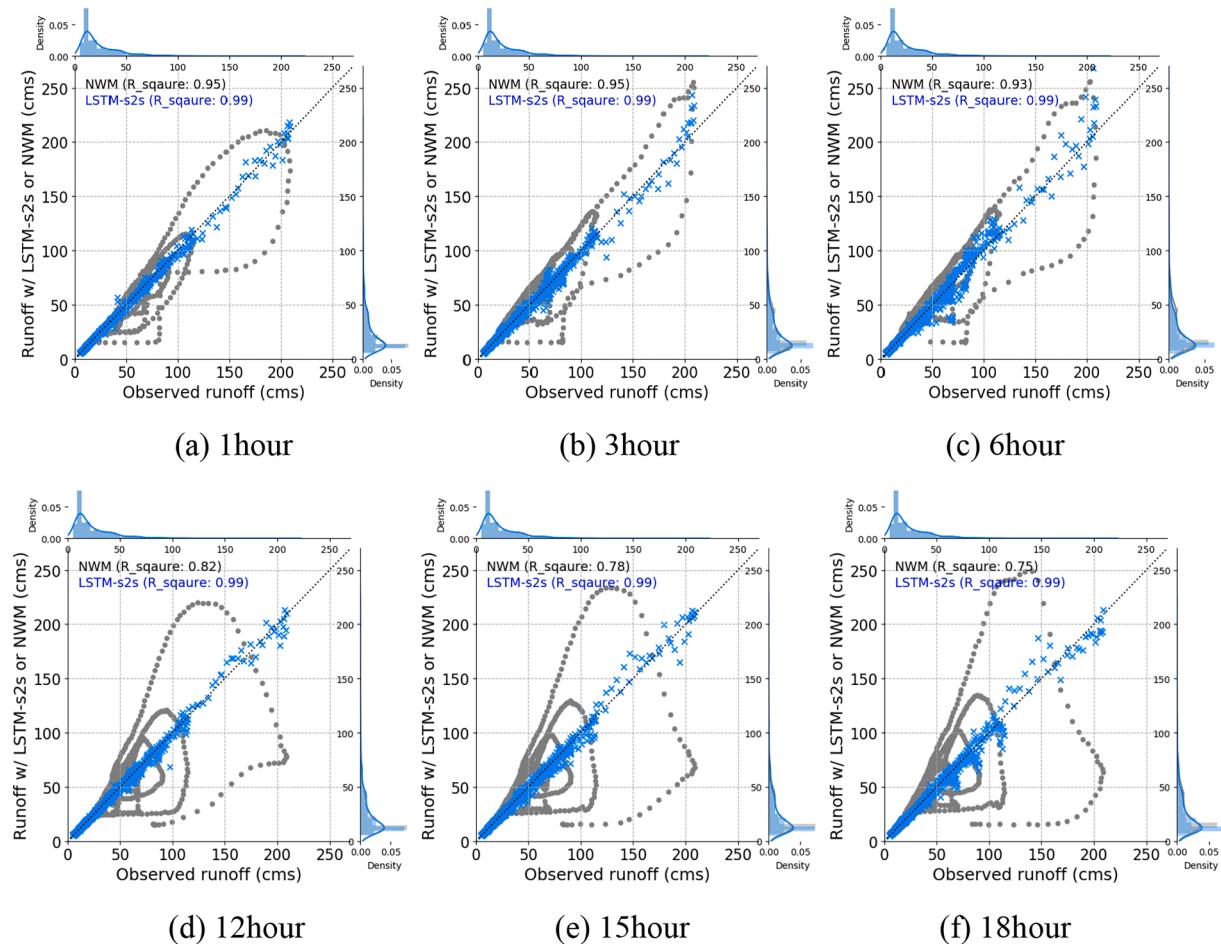


Fig. 9. Scatter plots comparing the forecasted runoff from NWM and runoff corrected by LSTM-s2s based post-processor framework. (a)-(e) represent scatter plots for 1 to 18 h.

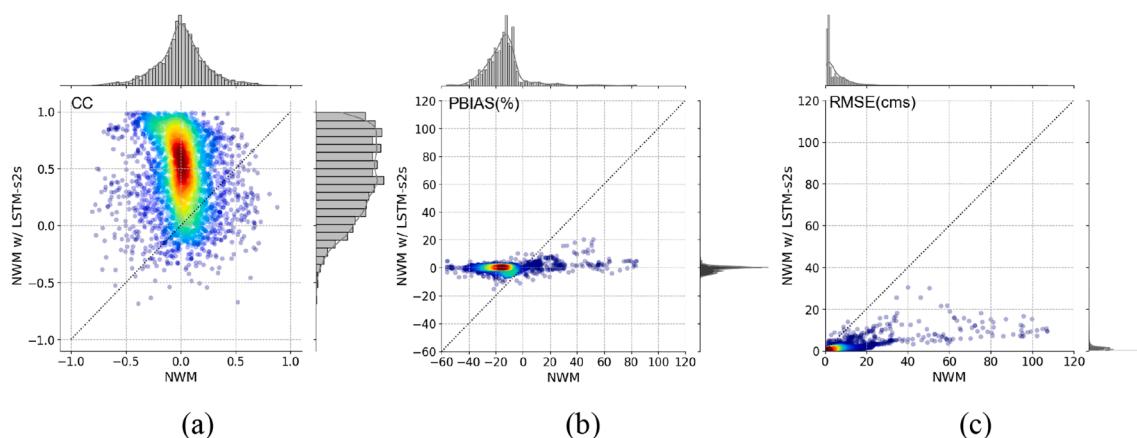


Fig. 10. Density scatter plots showing performance of two models (NWM, NWM w/ LSTM-s2s) using three metrics (CC, PBIAS, and RMSE).

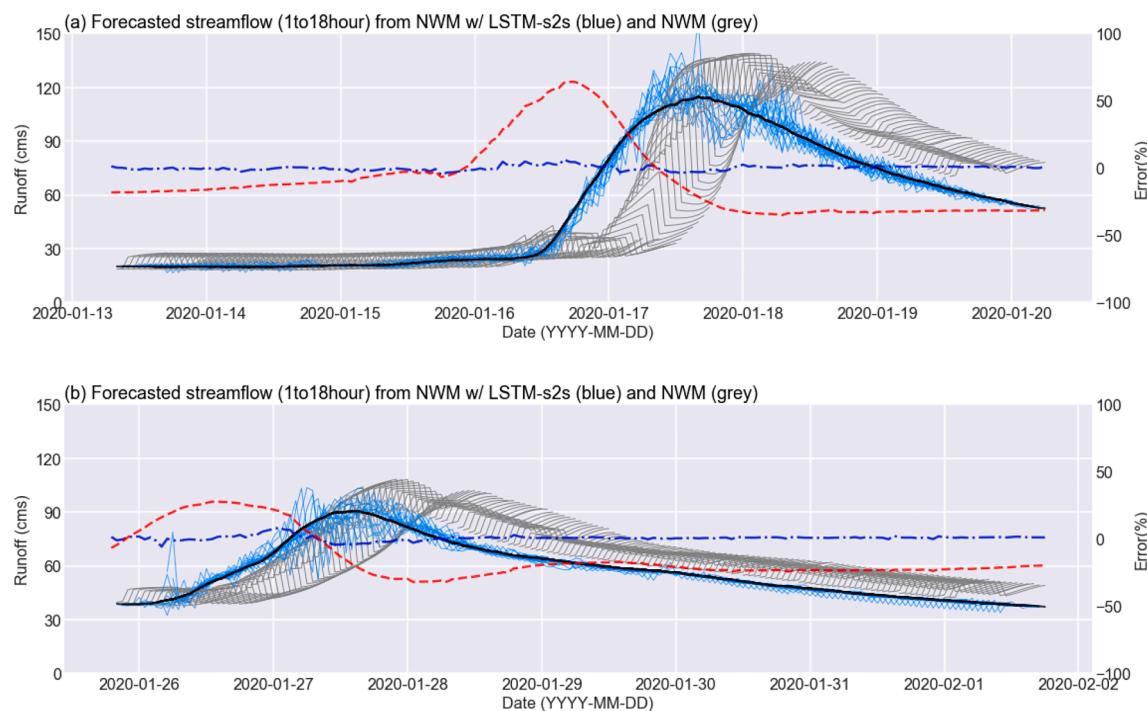


Fig. 11. Time-series of improved runoff from NWM with LSTM-s2s model (blue line) and NWM alone (grey line) for lead time between 1 and 18 h. Black line indicates observed runoff. Dark blue and red dot lines represent total errors in predicted runoff from NWM w/ LSTM-s2s and NWM models for 1–18 h. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

other regions. Although soil and topography data, such as soil properties, land use, and slope, were not considered in this study, they can be used to make efficient prediction results in other regions. In addition, this study considered only short-term prediction period between 1 and 18 h provided from NWM. The proposed model has the potential to be applied to medium- (to 10 days) and long-term (to 30 days) applications.

The error correction framework presented in this study can be applied to improve the performance of any hydrologic model and is not limited to the NWM. Since the framework works as a post-processor, it can predict new errors based on historic uncertainties that occur in a specific hydrological model. In addition, as this study has shown, the various factors related to the uncertainties in the outcomes can be used together depending on the characteristics of the local observation systems. Although only hourly-based datasets such as precipitation and runoff were used in this study, the error correction framework can consider multiple time-steps, ranging from sub-hourly to monthly for forecasting errors and improving models' performance.

5. Conclusions

In this study, we demonstrated how a LSTM-s2s model, which is a deep-learning model, can be used to improve the predictive performance of the NWM. The proposed method was applied in the Russian River basin, California, and the results indicated the following:

- 1) The performance of the NWM for forecasting runoff from 1 to 18 h was evaluated, and the prediction accuracy of the NWM decreased as lead times increased. It was shown that average metrics of CC, NSE, PBAIS, and RMSE were 0.65, 0.55, 10%, and 40 cms, respectively, for lead times between 12 and 18 h. In addition, the overall predictive performances of the NWM at three USGS stations were found to be lower compared to the actual runoff, which means that a post-processor could improve the prediction performance of the NWM.
- 2) Three factors, including previous time-step of input data, number of neurons, and batch size, effect on the performance of the LSTM-s2s model. The result indicates that the performance of the LSTM-s2s

decreased as sequence length and number of neurons increased. In addition, the change in batch size did not have a dramatic effect on simulation accuracy. The optimized model was based on 6 h for sequence length, 32 for number of neurons, and 128 for batch size, respectively.

- 3) The LSTM-s2s model obtained desirable results for error forecasting for each time step between 1 and 18 h with average values of CC of 0.95, NSE of 0.88, and PBIAS of -14%, respectively. The results of prediction errors from LSTM-s2s model were used for improving the predictive performance of the NWM.
- 4) It was found that using the LSTM-s2s as a post-processor significantly improved the predictive performance of the NWM for lead times of 1 to 18 h. Compared to the predicted runoff from the NWM, the NWM with the LSTM-s2s post-processor improved runoff predictions with R^2 values of 0.98 to 0.99. Moreover, it was found that LSTM-s2s based post-processor improved the predictive performance of NWM in terms of not only temporal pattern of runoff, but also the volume of predicted runoff.

Error correction through a post-processing framework is essential since many hydrologic models, such as the NWM, contain inherent uncertainties even after calibration. Therefore, we expect that the proposed post-process framework in this study can contribute to improving hydrological modeling performance by minimizing the uncertainty of various hydrological models besides the NWM. In addition, the suggested framework includes only upstream information, such as observed precipitation and errors as input data. Thus, it may be easy to apply the post-processor framework to improve the model's predictive ability for numerous regions. It is expected that the proposed framework can be applied for different basins and over 2.7 million river reaches around the United States for which the hydrological models (i.e., NWM) provides runoff predictions. This study also serves to highlight the power of deep learning models for solving hydrological problems.

CRediT authorship contribution statement

Heechan Han: Conceptualization, Methodology, Investigation, Visualization, Writing – original draft. **Ryan R. Morrison:** Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Abebe, A.J., Price, R.K., 2003. Managing uncertainty in hydrological models using complementary models. *Hydrol. Sci. J.* 48 (5), 679–692. <https://doi.org/10.1623/hysj.48.5.679.51450>.
- Adeyemi, O., Grove, I., Peets, S., Domun, Y., Norton, T., 2018. Dynamic neural network modelling of soil moisture content for predictive irrigation scheduling. *Sensors* 18 (10), 3408. <https://doi.org/10.3390/s18103408>.
- Agrawal, S., Barrington, L., Bromberg, C., Burge, J., Gazen, C., Hickey, J., 2019. Machine learning for precipitation nowcasting from radar images. arXiv preprint arXiv: 1912.12132.
- Arnaud, P., Lavabre, J., Fouchier, C., Diss, S., Javelle, P., 2011. Sensitivity of hydrological models to uncertainty in rainfall input. *Hydrol. Sci. J.* 56 (3), 397–410. <https://doi.org/10.1080/02626667.2011.563742>.
- Castelletti, A., Galelli, S., Restelli, M., Soncini-Sessa, R., 2010. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resour. Res.* 46 (9) <https://doi.org/10.1029/2009WR008898>.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
- Choi, C., Kim, J., Han, H., Han, D., Kim, H.S., 2020. Development of water level prediction models using machine learning in wetlands: A case study of upo wetland in South Korea. *Water* 12 (1), 93. <https://doi.org/10.3390/w12010093>.
- Datta, A.R., Bolisetti, T., 2016. Uncertainty analysis of a spatially-distributed hydrological model with rainfall multipliers. *Can. J. Civ. Eng.* 43 (12), 1062–1074. <https://doi.org/10.1139/cje-2015-0413>.
- Fan, H., Jiang, M., Xu, L., Zhu, H., Cheng, J., Jiang, J., 2020. Comparison of Long Short Term Memory Networks and the Hydrological Model in Runoff Simulation. *Water* 12 (1), 175. <https://doi.org/10.3390/w12010175>.
- Feng, Y., Cui, N., Hao, W., Gao, L., Gong, D., 2019. Estimation of soil temperature from meteorological data using different machine learning models. *Geoderma* 338, 67–77. <https://doi.org/10.1016/j.geoderma.2018.11.044>.
- J. Frame G. Nearing F. Kratzert M. Rahman Post processing the US National Water Model with a Long Short-Term Memory network 2020 <https://doi.org/10.31223/osf.io/4xhac>.
- Han, H., Kim, J., Chandrasekar, V., Choi, J., Lim, S., 2019. Modeling streamflow enhanced by precipitation from atmospheric river using the NOAA national water model: A case study of the Russian river basin for february 2004. *Atmosphere* 10 (8), 466. <https://doi.org/10.3390/atmos10080466>.
- Harr, M.E., 1989. Probabilistic estimates for multivariate analyses. *Appl. Math. Modell.* 13 (5), 313–318. [https://doi.org/10.1016/0307-904X\(89\)90075-9](https://doi.org/10.1016/0307-904X(89)90075-9).
- Haydon, S., Deletic, A., 2009. Model output uncertainty of a coupled pathogen indicator-hydrologic catchment model due to input data uncertainty. *Environ. Modell. Software* 24 (3), 322–328. <https://doi.org/10.1016/j.envsoft.2008.09.004>.
- 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>.
- Huang, Y., Chen, X., Li, Y.P., Huang, G.H., Liu, T., 2010. A fuzzy-based simulation method for modelling hydrological processes under uncertainty. *Hydrol. Processes* 24 (25), 3718–3732. <https://doi.org/10.1002/hyp.7790>.
- Hong, Y., Hsu, K.L., Moradkhani, H., Sorooshian, S., 2006. Uncertainty quantification of satellite precipitation estimation and Monte Carlo assessment of the error propagation into hydrologic response. *Water Resour. Res.* 42 (8) <https://doi.org/10.1029/2005WR004398>.
- Jin, X., Xu, C.Y., Zhang, Q., Singh, V.P., 2010. Parameter and modeling uncertainty simulated by GLUE and a formal Bayesian method for a conceptual hydrological model. *J. Hydrol.* 383 (3–4), 147–155. <https://doi.org/10.1016/j.jhydrol.2009.12.028>.
- Johnson LE, Hsu C., Zamora R., Cifelli R., 2016. Assessment and applications of distributed hydrologic model-Russian-Napa River Basins, CA. NOAA Technical Memorandum PSD-316, NOAA Printing Office, Silver Spring, MD, <https://doi.org/10.7289/V5M32SS9>.
- Kang, H., Sridhar, V., 2017. Combined statistical and spatially distributed hydrological model for evaluating future drought indices in Virginia. *J. Hydrol.: Reg Stud.* 12, 253–272. <https://doi.org/10.1016/j.ejrh.2017.06.003>.
- Kao, I.-F., Zhou, Y., Chang, L.-C., Chang, F.-J., 2020. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *J. Hydrol.* 583, 124631. <https://doi.org/10.1016/j.jhydrol.2020.124631>.
- Kim, J., Han, H., Johnson, L.E., Lim, S., Cifelli, R., 2019. Hybrid machine learning framework for hydrological assessment. *J. Hydrol.* 577, 123913. <https://doi.org/10.1016/j.jhydrol.2019.123913>.
- Kobold, M., Sušelj, K., 2005. Precipitation forecasts and their uncertainty as input into hydrological models. *Hydrol. Earth Syst. Sci.* 9 (4), 322–332. <https://doi.org/10.5194/hess-9-322-2005>.
- Kuczera, G., Parent, E., 1998. Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm. *J. Hydrol.* 211 (1–4), 69–85. [https://doi.org/10.1016/S0022-1694\(98\)00198-X](https://doi.org/10.1016/S0022-1694(98)00198-X).
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall-runoff modelling using long short-term memory (LSTM) networks. *Hydrol. Earth Syst. Sci.* 22 (11), 6005–6022. <https://doi.org/10.5194/hess-22-6005-2018>.
- Krzysztofowicz, R., 1999. Bayesian theory of probabilistic forecasting via deterministic hydrologic model. *Water Resour. Res.* 35 (9), 2739–2750. <https://doi.org/10.1029/1999WR900099>.
- Krzysztofowicz, R., 2001. The case for probabilistic forecasting in hydrology. *J. Hydrol.* 249 (1–4), 2–9. [https://doi.org/10.1016/S0022-1694\(01\)00420-6](https://doi.org/10.1016/S0022-1694(01)00420-6).
- Lee, H., McIntyre, N., Wheater, H., Young, A., 2005. Selection of conceptual models for regionalisation of the rainfall-runoff relationship. *J. Hydrol.* 312 (1–4), 125–147. <https://doi.org/10.1016/j.jhydrol.2005.02.016>.
- Li, L., Xia, J., Xu, C.Y., Singh, V.P., 2010. Evaluation of the subjective factors of the GLUE method and comparison with the formal Bayesian method in uncertainty assessment of hydrological models. *J. Hydrol.* 390 (3–4), 210–221. <https://doi.org/10.1016/j.jhydrol.2010.06.044>.
- Lin, G.F., Jhong, B.C., Chang, C.C., 2013. Development of an effective data-driven model for hourly typhoon rainfall forecasting. *J. Hydrol.* 495, 52–63. <https://doi.org/10.1016/j.jhydrol.2013.04.050>.
- Maskey, S., Guinot, V., 2003. Improved first-order second moment method for uncertainty estimation in flood forecasting. *Hydrol. Sci. J.* 48 (2), 183–196. <https://doi.org/10.1623/hysj.48.2.183.44692>.
- Maskey, S., Guinot, V., Price, R.K., 2004. Treatment of precipitation uncertainty in rainfall-runoff modelling: a fuzzy set approach. *Adv. Water Resour.* 27 (9), 889–898. <https://doi.org/10.1016/j.advwatres.2004.07.001>.
- McMillan, H., Jackson, B., Clark, M., Kavetski, D., Woods, R., 2011. Input uncertainty in hydrological models: an evaluation of error models for rainfall. *J. Hydrol.* 400 (1–2), 83–94.
- Melching, C.S., 1992. An improved first-order reliability approach for assessing uncertainties in hydrologic modeling. *J. Hydrol.* 132 (1–4), 157–177. [https://doi.org/10.1016/0022-1694\(92\)90177-W](https://doi.org/10.1016/0022-1694(92)90177-W).
- Montanari, A., 2007. What do we mean by ‘uncertainty’? The need for a consistent wording about uncertainty assessment in hydrology. *Hydrol. Processes: An Int. J.* 21 (6), 841–845.
- 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. *Trans. 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>.
- Muñoz, E., Tume, P., Ortíz, G., 2014. Uncertainty in rainfall input data in a conceptual water balance model: effects on outputs and implications for predictability. *Earth Sci. Res.* 18 (1), 69–75. <https://doi.org/10.1546/esrj.v18n1.38760>.
- Nearing, G., Sampson, A.K., Kratzert, F., Frame, J., 2020. Post-processing a Conceptual Rainfall-runoff Model with an LSTM. <https://doi.org/10.31223/osf.io/53te4>.
- Neitsch, S.L., Arnold, J.G., Kiniry, J.R., Williams, J.R., 2011. *Soil and Water Assessment Tool Theoretical Documentation Version 2009*. Texas Water Resources Institute, Temple, Texas.
- Ott, M., Su, Z., Schumann, A.H., Schultz, G.A., 1991. Development of a distributed hydrological model for flood forecasting and impact assessment of land-use change in the International Mosel river basin. *Proceedings of the Vienna Symposium. IAHS Pub* (No. 201).
- Ralph, F.M., Neiman, P.J., Wick, G.A., Gutman, S.I., Dettinger, M.D., Cayan, D.R., White, A.B., 2006/. Flooding on California’s Russian River: Role of atmospheric rivers. *Geophys. Res. Lett.* 33 (13) <https://doi.org/10.1029/2006GL026689>.
- Refsgaard, J.C., Storm, B., 1990. Construction and validation of hydrological models. In: *Distributed hydrological modelling*, Distributed Hydrological Modelling. Water Sci. Technol. Lib. 22, Springer, Dordrecht. https://doi.org/10.1007/978-94-009-0257-2_3.
- Rosenblueth, E., 1975. Point estimates for probability moments. *Proc. Natl. Acad. Sci.* 72 (10), 3812–3814. <https://doi.org/10.1073/pnas.72.10.3812>.
- Sahoo, S., Russo, T.A., Elliott, J., Foster, I., 2017. Machine learning algorithms for modeling groundwater level changes in agricultural regions of the US. *Water Resour. Res.* 53 (5), 3878–3895. <https://doi.org/10.1002/2016WR019933>.

- Shrestha, D.L., Solomatine, D.P., 2008. Data-driven approaches for estimating uncertainty in rainfall-runoff modelling. *Int. J. River Basin Manage.* 6 (2), 109–122. <https://doi.org/10.1080/15715124.2008.9635341>.
- Shrestha, D. L., Solomatine, D. P., 2009. Assessing uncertainty in rainfall-runoff models: Application of data-driven models. In: *Flood Risk Manage.* : Res. Pract. CRC Press London, UK, 1563-1573.
- Solomatine, D.P., Ostfeld, A., 2008. Data-driven modelling: some past experiences and new approaches. *J. Hydroinf.* 10 (1), 3–22. <https://doi.org/10.2166/ hydro.2008.015>.
- Soltan, H., Liao, H., Sak, H., 2016. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. *arXiv preprint arXiv:1610.09975*.
- Sønderby, C. K., Espeholt, L., Heek, J., Dehghani, M., Oliver, A., Salimans, T., Agrawal, S., Hickey, J., Kalchbrenner, N., 2020. MetNet: A Neural Weather Model for Precipitation Forecasting. *arXiv preprint arXiv:2003.12140*.
- Tung, Y.K., 2011. Uncertainty and reliability analysis in water resources engineering. *J. Contemp. Water. Res. and Educ.* 103 (1), 4.
- Vrugt, J.A., Ter Braak, C.J., Clark, M.P., Hyman, J.M., Robinson, B.A., 2008. Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation. *Water Resour. Res.* 44 (12) <https://doi.org/10.1029/2007WR006720>.
- Wang, J., Shi, P., Jiang, P., Hu, J., Qu, S., Chen, X., Chen, Y., Dai, Y., Xiao, Z., 2017. Application of BP neural network algorithm in traditional hydrological model for flood forecasting. *Water* 9 (1), 48. <https://doi.org/10.3390/w9010048>.
- Wu, R., Yang, L., Chen, C., Ahmad, S., Dascalescu, S.M., Harris Jr, F.C., 2018. Modeling error learning based post-processor framework for hydrologic models accuracy improvement. *Geosci. Model Dev. Discuss.* 1 <https://doi.org/10.5194/gmd-2018-136>.
- Wu, Z.Y., Lu, G.H., Wen, L., Lin, C.A., 2011. Reconstructing and analyzing China's fifty-nine year (1951–2009) drought history using hydrological model simulation. *Hydrol. Earth Syst. Sci.* 15 (9), 2881–2894. <https://doi.org/10.5194/hess-15-2881-2011>.
- Xiang, Z., Yan, J., Demir, I., 2020. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resour. Res.* 56(1), e2019WR025326. <https://doi.org/10.1029/2019WR025326>.
- Yilmaz, A.G., Muttil, N., 2014. Runoff estimation by machine learning methods and application to the Euphrates Basin in Turkey. *J. Hydrol. Eng.* 19 (5), 1015–1025. [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0000869](https://doi.org/10.1061/(ASCE)HE.1943-5584.0000869).