2022

# Variational Bayesian dropout with a Gaussian prior for recurrent neural networks application in rainfall–runoff modeling

S. Sadeghi Tabas

S. Samadi

ENVIRONMENTAL RESEARCH
LETTERS

LETTER • OPEN ACCESS

# Variational Bayesian dropout with a Gaussian prior for recurrent neural networks application in rainfall–runoff modeling

To cite this article: S Sadeghi Tabas and S Samadi 2022 *Environ. Res. Lett.* **17** 065012

ENVIRONMENTAL RESEARCH
LETTERS

OPEN ACCESS

RECEIVED
4 June 2021

REVISED
19 May 2022

ACCEPTED FOR PUBLICATION
23 May 2022

PUBLISHED
7 June 2022



LETTER

# Variational Bayesian dropout with a Gaussian prior for recurrent neural networks application in rainfall–runoff modeling

S Sadeghi Tabas[1] and S Samadi[2,*]

[1] The Glenn Department of Civil Engineering, Clemson University, Clemson, SC, United States of America
[2] Department of Agricultural Sciences, Clemson University, Clemson, SC, United States of America
* Author to whom any correspondence should be addressed.

**E-mail:** samadi@clemson.edu

**Keywords:** rainfall–runoff simulation, RNNs, uncertainty analysis, variational Bayesian dropout, Gaussian noise function, coastal plain hydrologic system

## Abstract

Recurrent neural networks (RNNs) are a class of artificial neural networks capable of learning complicated nonlinear relationships and functions from a set of data. Catchment scale daily rainfall–runoff relationship is a nonlinear and sequential process that can potentially benefit from these intelligent algorithms. However, RNNs are perceived as being difficult to parameterize, thus translating into significant epistemic (lack of knowledge about a physical system) and aleatory (inherent randomness in a physical system) uncertainties in modeling. The current study investigates a variational Bayesian dropout (or Monte Carlo dropout (MC-dropout)) as a diagnostic approach to the RNNs evaluation that is able to learn a mapping function and account for data and model uncertainty. MC-dropout uncertainty technique is coupled with three different RNN networks, i.e. vanilla RNN, long short-term memory (LSTM), and gated recurrent unit (GRU) to approximate Bayesian inference in a deep Gaussian noise process and quantify both epistemic and aleatory uncertainties in daily rainfall–runoff simulation across a mixed urban and rural coastal catchment in North Carolina, USA. The variational Bayesian outcomes were then compared with the observed data as well as with a well-known Sacramento soil moisture accounting (SAC-SMA) model simulation results. Analysis suggested a considerable improvement in predictive log-likelihood using the MC-dropout technique with an inherent input data Gaussian noise term applied to the RNN layers to implicitly mitigate overfitting and simulate daily streamflow records. Our experiments on the three different RNN models across a broad range of simulation strategies demonstrated the superiority of LSTM and GRU approaches relative to the SAC-SMA conceptual hydrologic model.

## 1. Introduction

Recurrent neural networks (RNNs) achieve state-of-the art performance on a wide range of sequence prediction tasks such as time series data (e.g. Gal and Ghahramani 2016a). These approaches are a class of artificial neural networks with a feedback loop that allows the algorithm to build up memory to process arbitrary sequences of inputs-output. RNNs are particularly suitable for modeling dynamical systems as they operate on input information as well as a trace of previously acquired information (due to recurrent connections) allowing for direct processing of temporal dependencies (e.g. Mirikitani and Nikolaev 2010).

The three widely used RNN architectures are the vanilla RNN (or simple RNN), long short-term memory (LSTM), and gated recurrent unit (GRU). Vanilla RNN uses a trained backpropagation through time (BPTT; Werbos 1990) that may lead to vanishing gradient problem for longer sequences of data training. LSTM networks are designed to overcome this shortcoming by storing information for longer periods of time (Werbos 1990). LSTMs are comprised

of cells which contain gates responsible for learning which data in a given sequence should be kept and which data can be forgotten (Hochreiter and Schmidhuber 1997). To simplify LSTM algorithm, the GRU was recently introduced by Cho *et al* (2014) to merge the memory state and hidden state into a single hidden state and combine the input and forget gates into an update gate. Since GRUs have fewer parameters, convergence is achieved quicker than LSTMs; nevertheless, GRUs contain sufficient gates and states for long-term memory retention (e.g. Werbos 1990).

Despite significant progress in the development of RNNs, limited studies have been focused on the application of them in watershed modeling studies. Kratzert *et al* (2018) is among the first studies that modeled the rainfall–runoff process using the LSTM for the contiguous United States (CONUS) and compared the results with the well-known Sacramento soil moisture accounting (SAC-SMA) model. An excellent application of the LSTM, so-called entity-aware-LSTM, is proposed by Kratzert *et al* (2019) with a capability for intelligent learning of catchment similarities as a feature layer. More recently, Feng *et al* (2020), tested a flexible procedure based on the LSTM, namely data integration, to simulate discharge across the CONUS. Their result revealed that the LSTM performed well in mountainous or snow-dominated regions while it is less capable for the regions with low discharge volumes and inter-annual storage variability. Jiang *et al* (2020) proposed a deep learning (DL) architecture by identifying physical approaches such as temporal dynamic geoscientific models as RNN layers to enhance AI geoscientific awareness. Their illustrative case of runoff modeling across the conterminous US demonstrates that the physics-aware DL model has enhanced runoff prediction accuracy and robust transferability for inferring unobserved processes. In addition, Feng *et al* (2021) proposed a novel DL input-selection ensemble model that couples methods with different input options and integrate different datasets such as satellite-based soil moisture product to improve streamflow modeling robustness in data scares regions. Their proposed ML model proved to be the best available tool for ungauged basin applications. Concurrently, Rahmani *et al* (2021) developed an LSTM architecture as a basin-centric lumped daily stream water temperature model, which was trained over 118 data-rich basins with no major dams in the conterminous US. Their results indicated that strong relationships exist between basin-averaged forcing variables, catchment attributes, and stream water temperature that can be simulated by a single model trained by data on the continental scale.

Most of the aforementioned studies viewed RNNs as a deterministic function, and as a result direct optimization (without complexity control) of these algorithms may lead to mediocre results due to uncertainty. One reason for this is, the parameter (weight) estimation involves inversion of a nonlinear system (here catchment system) from noisy data which typically is ill-posed (e.g. Casdagil 1989, Haykin and Principe 1998). In this situation, noises might exist within observations and measurements that are referred to as data uncertainties (also called aleatoric uncertainty, see Kiureghian and Ditlevsen 2009). In addition, there are many situations where uncertainties arise from the RNN structure choice and model parameters. This is referred to as model uncertainty or epistemic uncertainty. The standard approach to tackling ill-posed problems (both aleatoric and epistemic uncertainties) is by means of applying Bayesian approaches to modeling process. To best of our knowledge, very few studies have addressed the uncertainties in the RNNs simulations. For example, Zhu *et al* (2021) investigated two strategies to couple an LSTM with Gaussian processes (GPs) for drought forecasting. They used LSTM to parameterize a GP as well as a Gaussian post-processor. Inspiring by Kendall and Gal (2017), Fang *et al* (2020) used Monte Carlo dropout (MC-dropout) to simulate soil moisture and the associated uncertainty across CONUS. Their result revealed that MC-dropout provides a good estimation of predictive error in reproducing soil moisture dynamics recorded by the soil moisture active passive mission.

The current study used vanilla RNN, LSTM, and GRU to predict daily rainfall–runoff time series of a nonlinear and complex coastal plain drainage system. The motivation to use these models is related to the fact that these networks are properly set up with slightly different to form a regression model for time series prediction problems. The layers in the RNN networks are utilized to model the relationship between rainfall–runoff using the Bayes information of the weights updated following a heuristic approach to adjust and update the number of iterations of the RNNs. The goal is to develop such practical variational Bayesian inference to reason about uncertainty in rainfall–runoff simulation. Here, we coupled RNNs with variational Bayesian inference that is applied before RNN's weight parameters, mathematically equivalent to an approximation to the probabilistic GP model (marginalized over its covariance function parameters; see Damianou and Lawrence 2013). This probabilistic view of RNN simulation offers confidence bounds for watershed simulation analysis that hydrologists would rely on to analyze the data and make reliable simulation. The RNNs results were eventually compared with the well-known SAC-SMA model to determine the degree of RNN simulation success compared to a conceptual rainfall–runoff transformation model. We would like to stress that no simplifying assumptions are made on the use of variational dropout inference, and that the

results derived are applicable to any network architecture that makes use of dropout exactly as it appears in rainfall–runoff applications. We showed that the dropout objective, in effect, minimizes the Kullback–Leibler (KL) divergence between an approximate distribution and the posterior of a GP (marginalized over its finite rank covariance function parameters).

The current study is organized as follows. In section 2, the study area and the data are discussed. This follows with a description of the RNN set up, variational Bayesian dropout approach, and the proposed RNNs design for the study region. Section 3 presents the results of streamflow simulation, parameter sensitivity, robustness and feature ranking, and uncertainties. Finally, the conclusions and future work are presented in section 4.

## 2. Methodology

### 2.1. Study area and data

The Cape Fear River Basin (CFRB) is the largest basin in North Carolina (NC), USA, with an area of >9000 square miles. Major tributaries of this basin include the Deep River, the Haw River, the Northeast Cape Fear River, the Black River, and the South River. These river systems converge toward the coastal region to form a 30 mile-long estuary before flowing into the Atlantic Ocean at the Cape Fear. The Cape Fear supplies water to some of the fastest-growing counties in the US; roughly one in five North Carolinians gets their drinking water from the Cape Fear, including residents of Greensboro, Fayetteville, and Wilmington. The focus of this study is on the northeast portion of CFRB which drains 1714.70 square miles (4441.1 $km^2$) of CFRB, receives about 53.8 in $year^{-1}$ of precipitation on average, has a wetness index of 593.72 and is about 19% forested (figure 1). The Northeast Cape Fear River rises about 1 mile southeast of Mount Olive, NC in Wayne County and approximately 10 mi (16 km) south of Goldsboro and then flows south to the Cape Fear River at Wilmington, NC. On its course, it flows past Albertson, Hallsville, and Chinquapin. In Pender County near the Atlantic coast, it passes along the west side of Angola Swamp and Holly Shelter Swamp. It joins the Cape Fear River on the north end of Wilmington, forming an estuary that emerges at Cape Fear. The lower 50 mile (80 km) of the river is tidal.

In this study, daily streamflow records were retrieved from Daymet gridded dataset (Gauge number: 02108000) at the Northeast CFRB near Chinquapin in NC during 1980–2014. The collected data contains catchment aggregated (lumped) meteorological forcing data and observed 24 h daily streamflow. All catchment attributes used in this study were derived from gridded data products (Addor *et al* 2017) that include precipitation, short wave downward radiation, maximum and minimum temperature, and humidity. We employed the Daymet dataset since it has the most accurate spatial resolution (1 km grid resolution) as a basis for calculating the catchment averages and all available meteorological input variables.

### 2.2. RNN methods

RNNs are sequence-based models for time series data prediction (e.g. Gal and Ghahramani 2016a). The model's input is a sequence of data where at each time step a simple neural network (RNN unit) is applied to a single dataset, as well as to the network's output from the previous time step. RNNs are powerful tools that showed excellent performance for many modeling and prediction tasks. Here we implemented vanilla RNN, LSTM, and GRU architectures. The training process of the three RNNs were adapted using BPTT, a procedure similar to the classical back propagation (BP; Werbos 1990). The training processes are as follows: First, the output values of hidden cells were calculated according to the forward calculation method. Second, the error values of hidden layer were calculated on the basis of BP of time step and network structure. Furthermore, the gradients of each weight were computed according to the corresponding error terms. Finally, the weight coefficients were updated by adaptive moment (Adam; Kingma and Ba 2014) estimation algorithm. Similar to a BP neural network, the number of iterations was one of the main criteria affecting the model performance. The mathematical functions of vanilla RNN, LSTM, and GRU architectures are compared with the SAC-SMA model that are explained with respect to their mathematical function in supporting information section).

### 2.3. Variational Bayesian dropout with a Gaussian prior

Bayesian probability theory offers mathematically grounded tools to reason about model uncertainty (e.g. Gal and Ghahramani 2016a). Indeed, by extending the mathematically grounded theory of RNNs with Bayesian theory, both epistemic and aleatoric uncertainties present in the data and the model can be captured. With this, not only comparable performance to current state-of-the-art results in rainfall–runoff simulation can be reached, but also the quality of the predictions can be assessed by their predictive uncertainty. Variational Bayesian methods are a family of techniques for approximating intractable integrals arising in Bayesian inference and machine learning. They are typically used in complex statistical models consisting of observed variables as well as unknown parameters and latent variables. As typical in Bayesian inference, the parameters and latent variables are grouped together as unobserved variables. Variational Bayesian methods are primarily used for two purposes: (a) to provide an analytical approximation to the posterior probability of the unobserved variables, in order to do statistical inference over these

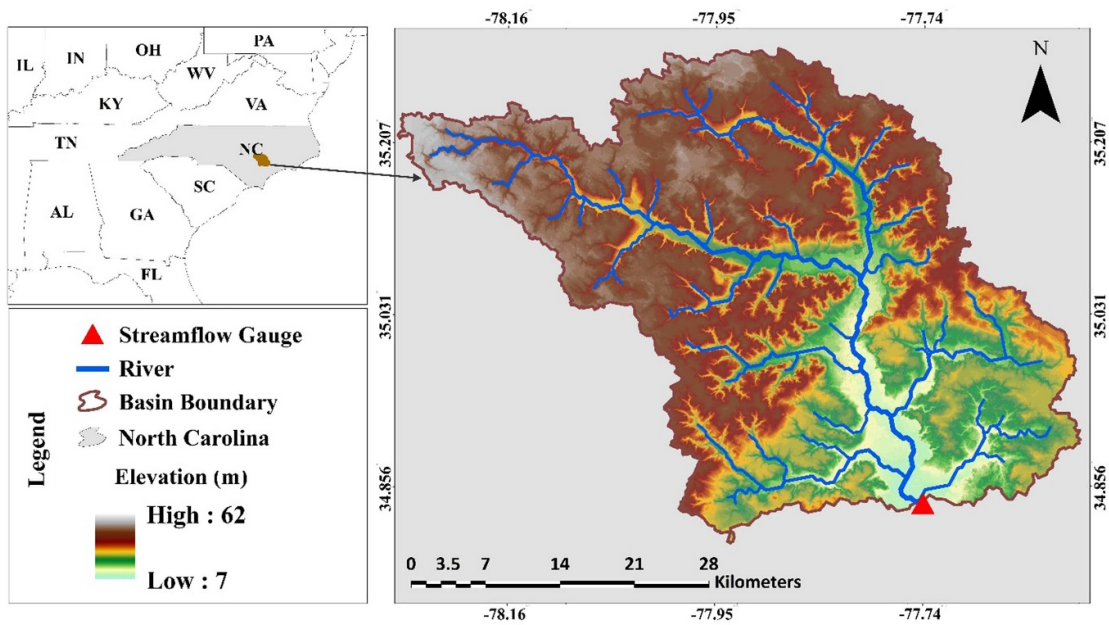


**Figure 1.** The Northeast CFRB located in NC with a slope of >50% in upland and a gentle slope in the downstream portion.

variables; (b) to derive a lower bound for the marginal likelihood of the observed data (i.e. the marginal probability of the data given the model, with marginalization performed over unobserved variables). This is typically used for performing model selection, the general idea being that a higher marginal likelihood for a given model indicates a better fit of the data by that model and hence a greater probability that the model in question was the one that generated the data. Variational Bayesian inference (or MC-dropout) is a practical approach for approximating inference in large and complicated models (Gal and Ghahramani 2016b). Here, we argue that the use of dropout (and its variants) in RNNs can be interpreted as a variational Bayesian approximation of a well-known probabilistic model: the GP as stressed by Rasmussen and Williams (2006). As mentioned in the introduction section, a GP is a distribution over functions fully specified by a mean and covariance function (marginalized over its covariance function parameters; see Damianou and Lawrence 2013). Thus, the posterior predictions of a GP are weighted averages of the observed data where the weighting is based on the covariance and mean functions.

There are two main types of uncertainties in Bayesian modeling: epistemic (model uncertainty) and aleatoric (data uncertainty). The intrinsic randomness of the data generation process is described by aleatoric uncertainty which cannot be explained by collecting further observations or data samples. This type of uncertainty is the result of unknowns that refers to the notion of randomness, that is, the variability in the outcome of modeling. Due to the model inputs, there is no way of finding the exact output of a concrete evaluation as there will be a certain amount of variance in the result. The aleatoric uncertainty has two types: homoscedastic and heteroscedastic (Kendall and Gal 2017). Homoscedastic uncertainty is invariant to different inputs, meaning it remains consistent regardless of inputs. Heteroscedastic uncertainty, on the other hand, changes over different inputs to a model. In other words, for some inputs, it could output more noisy results compared to when the other inputs are given. In order to capture the aleatoric uncertainty, we would have to tune the observation noise variance. As mentioned before, homoscedastic uncertainty will output the constant observation noise for all input data point and therefore the observation noise variance is constant for all input parameters, whereas in the case of heteroscedastic uncertainty it will vary according to their definition. In other words, in heteroscedastic uncertainty the observation noise variance depends on the input parameters and can be estimated as a model output. As a result, heteroscedastic models are helpful when some parts of the observation space might have higher noise levels than others (e.g. low flow values).

On the other hand, Epistemic uncertainty results from a lack of data that is potentially available, i.e. a larger number of observations is able to explain this kind of uncertainty. To capture epistemic uncertainty in the RNNs, we included a Gaussian prior distribution over its weights, $W \sim \mathcal{N}(0, \sigma)$ as a Bayesian neural network (BNN). BNN marginalizes RNNs' weight parameters by averaging over all possible weights and it replaces the deterministic RNNs' weight parameters with distributions. Due to Bayesian inference, given a dataset $X = \{x_1, \ldots, x_N\}$ (e.g. rainfall, temperature etc) and $Y = \{y_1, \ldots, y_N\}$ (e.g. streamflow), the posterior over weights

is $p(W|X, Y)$. The model likelihood would be $p(y|f^w(x))$ which $f^w(x)$ denotes the random output of the BNN. Assuming the likelihood as a Gaussian with $\{\mu : f^w(x)\}$ and $\{\sigma : \text{observation noise}\}$, we have $p(y|f^w(x)) = \mathcal{N}(f^w(x), \sigma^2)$. However, it is challenging to compute the exact posterior inference as the marginal probability $p(Y|X)$ cannot be assessed analytically, but it can be approximated. To do this, several approximations proposed by various studies (Graves 2011, Blundell *et al* 2015, Hernandez-Lobato *et al* 2016, Gal and Ghahramani 2016a). In these approximate inference techniques, the posterior $p(W|X, Y)$ is fitted to a simple distribution $q_\theta^*(W)$. This proposes an optimization task as an alternative to the intractable problem of averaging over all weights in the BNN. In other words, rather than optimizing the parameters of the original RNN, we considered to optimize the parameters of a simple distribution.

During the training process, dropout randomly drops some nodes to avoid them from too much co-tuning (see Gal and Ghahramani 2016a). During testing period, the dropout would be performed to generate random predictions by sampling from the approximate posterior. This approach is equivalent to finding a simple distribution $q_\theta^*(W)$ which minimizes the KL divergence to the true model posterior $p(W|X, Y)$. Based on Jordan *et al* (1999), the minimization objective is presented as follows:

$$\mathcal{L}(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i|f^{\hat{W}_i}(x_i)) + \frac{1-p}{2N}||\theta||^2 \tag{1}$$

where, $N$ and $p$ denote the number of data points and dropout probability respectively, $i$ is the sampled masked model weights $\widehat{W}_i \sim q_\theta^*(W)$, and $\theta$ is a set of simple distribution parameters that need to be optimized. For a Gaussian likelihood, the negative log likelihood can be further simplified (Kendall and Gal 2017):

$$-\log p\left(y_i \middle| f^{\hat{W}_i}(x_i)\right) \propto \frac{1}{2\sigma^2}\left\|y_i - f^{\hat{W}_i}(x_i)\right\|^2 + \frac{1}{2}\log \sigma^2. \tag{2}$$

As a result, the predictive variance can be approximated as:

$$\text{Var}(y) \approx \sigma^2 + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{W}_t}(x)^T f^{\hat{W}_t}(x_t) - E(y)^T E(y) \tag{3}$$

where $E(y) \approx \frac{1}{T}\sum_{t=1}^{T} f^{\hat{W}_t}(x)$. The term $\sigma^2$ in the predictive uncertainty corresponds to the intrinsic noise in the data, while the second term explains the model parameter uncertainty. To capture the aleatoric uncertainty, the observation noise parameter $\sigma$ should be tuned. As we mentioned above, homoscedastic aleatoric uncertainty assumes the observation noise is constant for every input data. However, it can vary with input in heteroscedastic aleatoric uncertainty. Thus, we decided to perform the heteroscedastic (rather than homoscedastic) model uncertainty assessment as it is important once some inputs or data potentially having more noisy outputs than others (e.g. low flow values), so we can make it data-dependent and estimate it as a function of data:

$$\mathcal{L}_{\text{RNN}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\sigma(x_i)^2}||y_i - \text{f}(x_i)||^2 + \frac{1}{2}\log \sigma(x_i)^2. \tag{4}$$

To capture both epistemic and heteroscedastic aleatoric uncertainties, we turn heteroscedastic RNN (equation (4)) into a Bayesian RNN by placing a distribution over its weights. To do so, we draw model weights from approximate posterior $\widehat{W}_i \sim q(W)$ to compute a model output including predictive mean and variance $\left\{(\hat{y}, \hat{\sigma}^2) = f^{\hat{W}}(x)\right\}$. As a result, the minimization objective induces as follows:

$$\mathcal{L}_{\text{BRNN}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\hat{\sigma}_i^2}||y_i - \hat{y}_i||^2 + \frac{1}{2}\log \hat{\sigma}_i^2. \tag{5}$$

For numerical stability, we train the network to predict the log variance, $s = \log \hat{\sigma}_i^2$, for $\sigma_x^2$, as the loss avoids a potential division by zero. In this procedure $s$ is implicitly learned by the regression task. We also applied an exponential mapping to regress unconstrained scalar values and $\exp(-s_t)$ would have a positive domain which is valid as a variance.

$$\mathcal{L}_{\text{BRNN}}(\theta) = \frac{1}{2N}\sum_{i=1}^{N} \left[\exp(-s_i)(y_i - \hat{y}_i)^2 + s_i\right]. \tag{6}$$

The term $s_i$ plays a regularization role to prevent unreservedly decreasing of the $\exp(-s_i)$ during the training. To summarize, the predictive uncertainty in the combined model can be approximated as follows:

$$\text{Var}(y) \approx \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t^2 - \left(\frac{1}{T}\sum_{t=1}^{T} \hat{y}_t\right)^2 + \frac{1}{T}\sum_{t=1}^{T} \hat{\sigma}_t^2 \tag{7}$$

where $\{\hat{y}_t, \hat{\sigma}_t^2\}_{t=1}^{T}$ denotes a set of $T$ sampled outputs: $\hat{y}_t, \hat{\sigma}_t^2 = f^{\hat{W}_t}(x)$ for randomly masked weights $\hat{W}_t \sim q(W)$. The performance criteria comparing different RNN-Bayesian dropout is presented in supporting information section.

### 2.4. RNNs design for the Northeast Cape Fear River Basin

We developed streamflow simulation for the Northeast CFRB based on the theory of a conceptual hydrologic model combined with RNNs. As illustrated in

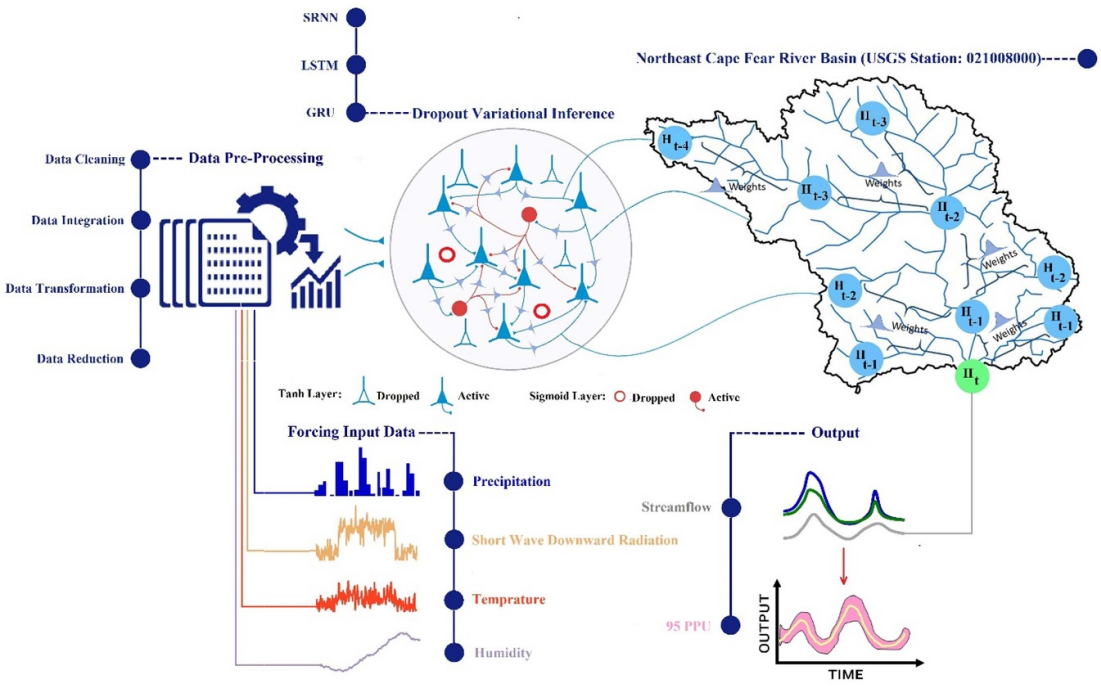


**Figure 2.** The proposed RNN workflow and structure for daily streamflow simulation across the Northeast Cape Fear Basin.

figure 2, various RNNs were constructed and coupled with a variational Bayesian approach. We used a one-layer RNN network that contained a cell/hidden state length ranging from 1 to 100. We added dropout approach ranging from 10% to 70%. The hyperparameters include sequence length of meteorological data map series, batch size, and the optimizer that compared with the learning curve. To fully capture the spatio-temporal variability and dynamics of streamflow data patterns, we decided to keep the sequence length of data constant at 365 d (indicates a complete water year). In the traditional hydrologic modeling approach, the number of iteration steps defines the total number of model evaluations performed during calibration (given an optimization algorithm without a convergence criterion). The corresponding term for RNNs is so-called epoch. One epoch is defined as the period in which each training sample is used once to update the model parameters (e.g. if the dataset contains of 1000 training samples and the batch-size is 10, one epoch would equal to 100 iterations; number of training samples divided by the number of samples per batch). In each iteration, 10 of the 1000 samples were taken without replacement until all 1000 samples were used once. This makes, each time-step of the discharge data to be simulated exactly once. This is similar to one iteration in a traditional hydrologic model calibration, with a significant difference of generating every sample independently (e.g. Kratzert *et al* 2018). To reduce anomalies in the data, all input features (the meteorological forcing variables) as well as the output (the discharge) data were normalized by subtracting the mean and dividing by the standard deviation (Minns and Hall 1996, LeCun *et al* 2012). The mean and standard deviation were calculated only for the calibration period. In the test period, the output was retransformed using the normalization parameters obtained from the calibration period.

This study divided the streamflow data into three subsets, referred to as training, validation, and test datasets. The training period is so important to get a good actual result because a small division of dataset in training period can lead to (a) a much shorter period of data that is used for the actual weight updates and (b) a high risk of overfitting to the short validation period (Kratzert *et al* 2018). In addition, RNNs with a low number of hidden units are quite sensitive to the initialization of their weights. It is thus recommended to repeat the calibration task several times with different random seeds and select the best performing model realization (Bengio 2012). We used the first 20 years of the 30 year calibration period as training data (01 January 1980 to 31 December 1999) while the last 10 years for validation and hyperparameter tuning (01 January 2000 to 31 December 2009). The first two subsets are used to derive the networks parametrization (calibration in the context of hydrologic simulation) and the remainder of data to diagnose the actual performance (validation in the context of hydrologic simulation).

## 3. Results and discussion

Similar to continuous conceptual hydrological models, we used meteorological forcing data to update

a number of values in the internal cell states. In addition, there are cell states in the RNN networks which can be interpreted as storage capacity often used in hydrologic modeling setup. Updating of internal cell states (or storages) is regulated through a number of gates: the first gate regulates the storages depletion, the second one regulates storage fluctuations, and the third gate regulates the storages outflow (especially for the LSTM and GRU networks). Each of these gates includes a set of adjustable parameters that are exploited during the calibration. Updates of the cell states, during the validation period, only rely on the input at a specific time-step and the states of the last time-step (given the learned parameters of the calibration period). Unlike hydrological models, RNNs does not 'know' the law of mass conservation and the equations governing storage and movement of water within the principal compartments of a drainage system (e.g. evapotranspiration (ET) or infiltration/percolation, runoff, etc.). RNN networks must learn these physical phenomenon and laws purely from the data structure and patterns. We performed a trial-and-error process to tune hyperparameters and derive the best values. We first begin presenting our results by presenting hyperparameter tuning procedure and uncertainty assessment. This is followed by the analysis of the results, demonstrated in the following sections.

### 3.1. Hyperparameter tuning

To get the best simulation result, hyperparameter values need to be tuned for each network. We performed a trial-and-error process to tune the parameters and derive the best hyperparameter values. The tuning result of some hyperparameters includes sequence length of streamflow data map series, batch size, and the optimizer. Among three RNNs, vanilla RNN has the least dependencies with only short-term dependencies of 10 or less time steps. The reason is that this network has vanishing or exploding gradients issues which reveals itself in an error signal during the backward pass of the network training that either grows against infinity or diminishes towards zero. This thus limit vanilla RNN ability to learn long-term temporal dependences. However, when we look at rainfall–runoff relationships at a catchment scale, there are different processes that are interconnected. These relationships including evapotranspiration, the dynamics of depletion and replenishment of water storage in a lowland area, etc that are highly nonlinear and extremely complex as they are interrelated to various sub-processes involved in a hydrologic cycle. Therefore, the interdependencies of rainfall–runoff processes are well above 10 d (ten-time step in our daily streamflow simulation) as stated elsewhere (see Kratzert *et al* 2018). To ensure a high-precision prediction result, we assumed that the number of training epochs and the number of hidden layers ranging from 1 to 100 for all networks. All other boundary conditions of modelling e.g. input data (meteorological forcing data), batch size, and the number of one input sequence were kept identical.

In this study, we first explored the influence of number of iterations and hidden cells on the simulation's accuracy. The performances of different parameters and recurrent networks are shown in figures 3 and 4 for calibration and validation periods, respectively. The results suggested that the accuracy of RNN simulations enhanced when the number of iterations increased over time. It is interesting to note that, an increase in the number of maximum iterations does not significantly enhance model precision when the number of maximum iterations reaches a certain limit. The same results can be interpreted for the effect of hidden cells numbers on the model performance. Here, we assumed that BPTT training method uses forward and backward algorithms to calculate the output value and the error of each hidden cell, respectively. The network weights were continuously updated to reduce errors and bias in simulation.

Initial weights of the network were set as random values, and a gradient-based Adam optimizer was used to adjust the network weights. After the model training reaches a certain limit, increasing the number of iterations was no longer significant for the model improvement (model met the convergence criteria). The influence of hidden neurons on model precision has always been a key problem in the RNN setting because they can influence the error on the nodes to which their output is connected. The minimal error reflects better network stability, while higher error reflects worst stability. The optimal number of hidden neurons can be estimated by a trial-and-error method which costs time but is more efficient to reach high accuracy.

With respect to each model performance, the vanilla RNN network performed a few simulations with acceptable precision (NSE $>$ 0.7) compared to the LSTM and GRU models. Figure 3 shows that the performance of LSTM model is significantly influenced by the number of iterations; thereby an increase in the number of hidden neurons did not significantly improve model precision when the number of hidden cells exceeded 30. Although, the influence of hidden nodes on the accuracy of the GRU simulation is not obvious for the results, our analysis showed that if the number of hidden nodes is less than 20, the GRU network requires more iterations to converge and ensure precise simulation (see figure 3). Among the three networks, GRU required less hidden layers which conveys the fact that the convergence of the GRU network is quicker than the other two. Additionally, reasonable values of other criteria (KGE, TRMSE and ROCE) revealed that the GRU simulations was satisfied the full suite of high flow, low flow, and water balance objectives against other algorithms. The high flows are more representative of direct runoff in a

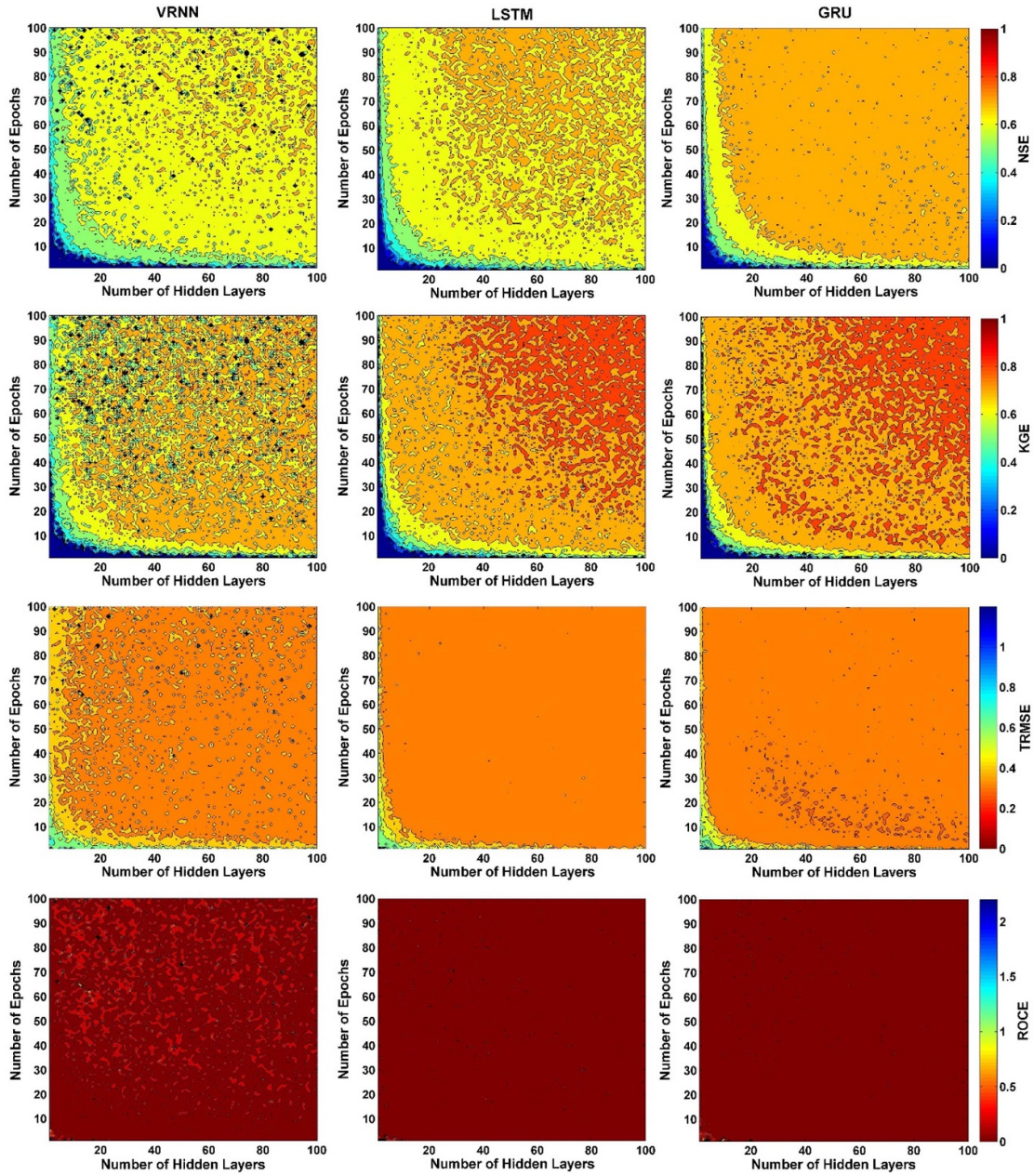


**Figure 3.** Performance surface diagram of different RNNs in the calibration period. One epoch is equal to one forward pass and one backward pass of all the training data. To find optimal number of epochs, the performance metrics of train and test data were calculated for each epoch with different number of maximum epochs and hidden cells. This prevents the network from overfitting and provided an approximated range of epochs to start with.

river system while the low flow related to long-term sustainability of streamflow records that controlled by the interaction of baseflow with riparian ET during extended dry periods. Therefore, during periods of low precipitation and high PET (low flow periods), the water draining to the river system may be limited and that maybe lost to ET; nevertheless, GRU did quite a good job by adequately simulating the baseflow events.

In order to evaluate the actual performance of various RNN networks, we ran the trained models for 4 year of the test period (01 January 2010 to 31 December 2013). As illustrated in figure 4, vanilla RNN reached an unreliable result during the test period. In fact, the vanilla RNN was unable to perform well for a long sequence because of the gradient vanishing/exploding problem. Contrary to the vanilla RNN, LSTM and GRU allowed the model to preserve the sequence information; thereby performed better during testing period. Although LSTM proved to be robust when compared with the vanilla RNN, its computational complexity is a burden due to additional weight matrices. The GRU recurrent structure reduced these computational complexities of LSTM by combining the input and forget gates of LSTM into a single update gate and combining the hidden and cell states into a single hidden state. Compared to the LSTM and vanilla RNN, GRU proved to be

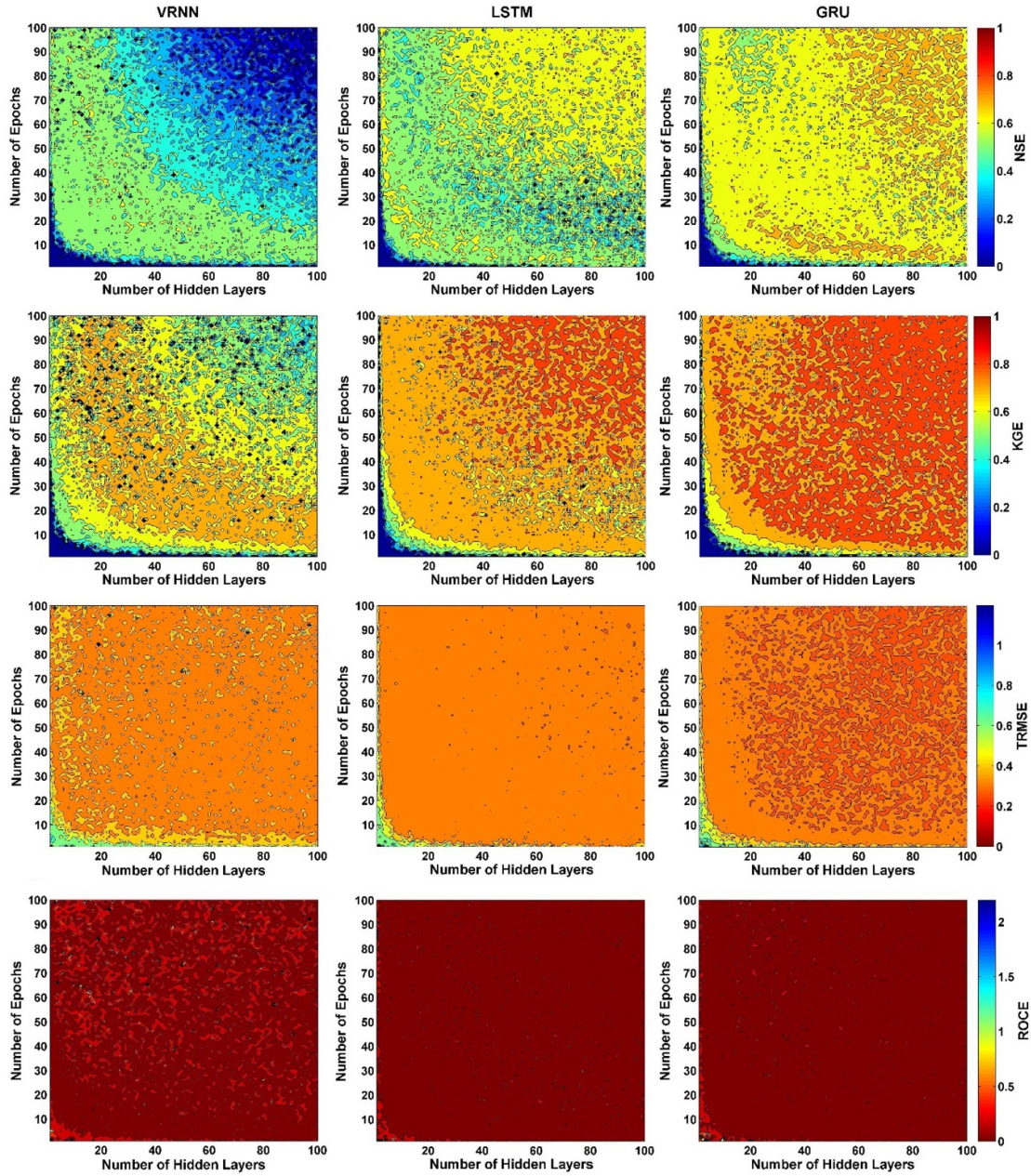


**Figure 4.** Performance surface diagram of different RNNs in the validation period.

superior in accounting for the fluctuations in a time series data and understanding of catchment processes purely from data patterns that control its hydrological input-state-output behavior. More specifically, the GRU algorithm achieved an average of 65% accuracy for the NSE value as evident in the top-right corner (red surface area) of the performance optimization surface in figure 4 with a number of hidden cells and number of epochs >20.

We believe that the effect of maximum iterations on model precision is crucial and should be given more attention when establishing these RNN networks. Nevertheless, the effect of hidden nodes on model precision is less significant. However, in the process of training streamflow data, RNN algorithms automatically learned the hidden information in the data and made it more resilient to errors and outliers. As the amount of streamflow records increased, the prediction accuracy of these networks was further improved. Most importantly, a sequence of hidden states and a sequence of predictions, which determine the input-to-hidden weight matrix, defined how much importance to accord to both the present input and the past hidden state. In this study, the error that the present input and the past hidden state generated returned via backpropagation that used to adjust their weights until error did not go any lower.

In order to assess the robustness of networks and interpret the networks performance distinctly, the probability density function (PDF) of various performance metrics compared during calibration and validation periods. As illustrated in figure 5, the NSE

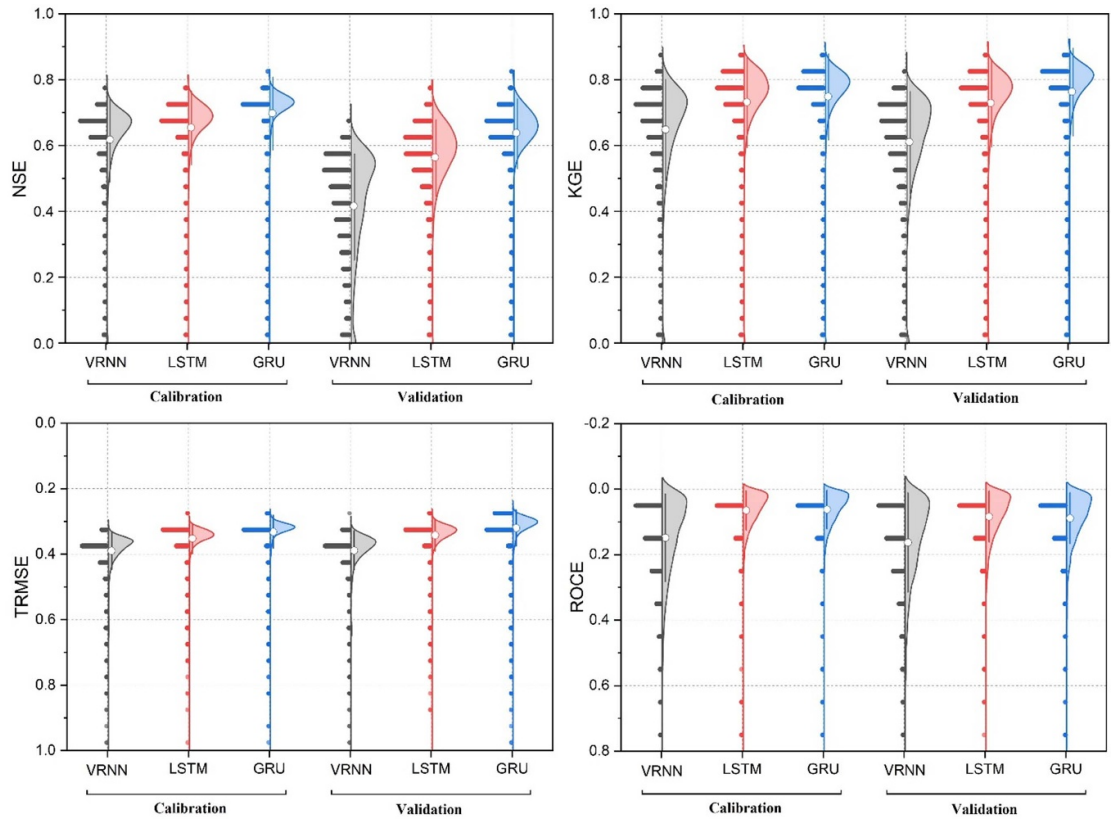


**Figure 5.** PDFs for various metrics during calibration and validation periods.

of vanilla RNN, LSTM and GRU are, respectively 0.61, 0.65 and 0.72 during calibration period. But the corresponding values in the validation periods are 0.41, 0.58 and 0.64.

### 3.2. Variational Bayesian uncertainty assessment

RNNs with network weights were treated as random variables with an appropriate defined likelihood function. Approximating the posterior distribution over the weight matrices with a Gaussian prior (with one component fixed at zero and small variances) led to a flexible optimization objective. Optimizing this objective was identical to performing a new variant of MC-dropout in the respective RNNs. In the new MC-dropout variant, we repeated the traditional dropout mask (naïve dropout) at each time step for both inputs, outputs, and recurrent layers (drop the same network units at each time step). This is in contrast to the existing MC-dropout techniques where different dropout masks are sampled at each time step for the inputs and outputs alone (no dropout is used with the recurrent connections).

We first performed MC-dropout with predefined dropout rates and tested various rates afterwards. We then noticed that although model diversity can be increased with higher dropout rates, it can come at a cost of reduced accuracy, which was undesirable. Therefore, it seems there is a fundamental trade-off between model diversity and the accuracy of individual models in the MC-dropout. Thus, we treated dropout rate as an hyperparameter chosen based on the maximum likelihood estimation (MLE) for the validation data. At the testing period (inference), dropout was activated to allow randomly sampling from the approximate posterior (stochastic forward passes; referred to as MC-dropout). In our case, dropout rate of $\approx$0.3 was an optimal value for accurate daily streamflow predictions and uncertainty estimation. The probability plots for aleatoric uncertainty (inherent noise in data), epistemic uncertainty (model structure uncertainty), and the total uncertainty are presented in figure 6. The probability plots visually showed the quality of each uncertainty-estimating component. To estimate the uncertainty, we followed different simulations strategies. For example, we ran pre-trained models 200 times to generate 200 simulation time series (based on the weighted posterior distribution) and then used those 200 simulations to compute both epistemic and aleatory uncertainties.

An ideal uncertainty estimate would produce a CDF that is identical to a 1:1 plot (black dash-lines). As illustrated in figure 6(a), if we only consider inherent noise, the uncertainty would be over-estimated in both GRU and LSTM (figure 6(a)). Considering only the aleatory term of uncertainty is not enough

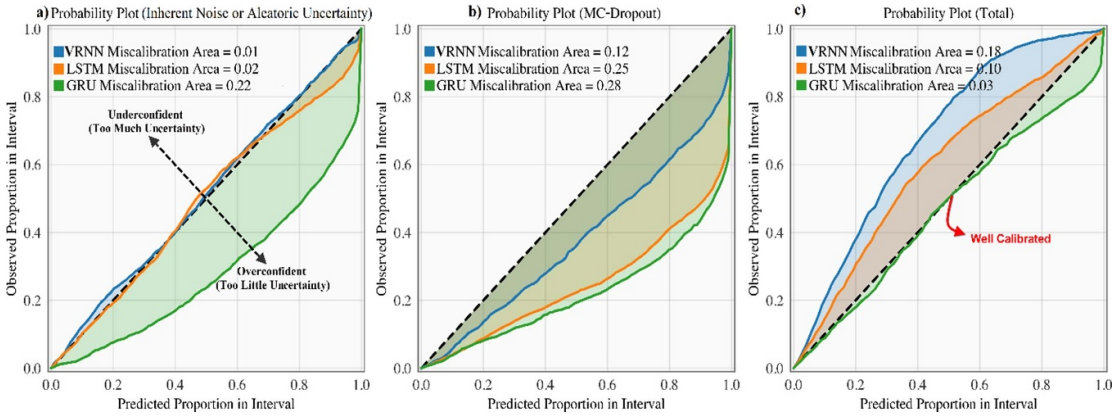


**Figure 6.** Calibration performance plots of the RNNs with respect to three uncertainty estimates: (a) data noise ($\sigma^2_{x;\text{ inheret noise or aleatoric uncertainty}}$), (b) network weight uncertainty ($\sigma^2_{\text{MC-dropout}}$), and (c) combined uncertainty ($\sigma^2_{\text{comb}}$).

for quantifying the overall uncertainty. As we mentioned, epistemic uncertainty refers to the deficiencies by a lack of knowledge or data. Therefore, the accuracy of models that calibrated based on insufficient data (which cannot represent the whole features and patterns contributing to rainfall–runoff process), would be decreased during the inference stage consequently increases the uncertainty in simulations. On the other hand, the variational Bayesian inference or MC-dropout lies below the 1:1 line toward the right end, which means that the model is overconfident, and the predictive errors occurred less than anticipated (figure 6(b)). Hence, the pattern indicates that MC-dropout alone over-estimated the uncertainty toward the error range (too little uncertainty). However, the combination of epistemic and aleatoric uncertainty $\left(\sigma^2_{\text{comb}} = \sigma^2_x + \sigma^2_{\text{MC-dropout}}\right)$ of the GRU simulation was closer to the one-to-one line than any other network. Thus, we perform a procedure to address both issues that contribute to simulation uncertainty which are random noise inside the measurements and the uncertainty that comes from the lack of the data which is resulted in model parameters tuning.

In the LSTM and vanilla RNNs simulations, there was a slightly larger gap between $\sigma^2_{\text{comb}}$ and the 1:1 line than $\sigma^2_x$, but it was similar to the GRU regarding the $\sigma^2_{\text{MC-dropout}}$. Variational Bayesian inference can improve performance by implicitly learning attenuation of systematic noise in the data.

When Gaussian random noise was included to the models, estimated uncertainties increased significantly. This reflects the fact that the proposed aleatory data noise scheme effectively estimated random noise (see figure 6(c)). It is interesting to note that both LSTM and vanilla RNN were unable to formulate and predict uncertainty during high flow records, although the resulting predictive uncertainty may depend heavily on the non-linearity in data and model prior. The standard deviation of variational Bayesian can be mitigated by adding more supervised data. Based on these experiments, it seems uncertainty estimation results have a statistically significant improvement for all three RNN simulations. Although, GRU was more efficient in capturing the total uncertainty (the combination of epistemic and aleatoric terms) in predictions (close to the 1:1 dash-line) while vanilla RNN and LSTM underestimated the uncertainty and showed under-confident results. These results suggested that it is important to address model predictive uncertainty using $\sigma^2_{\text{comb}}$, whereas using either $\sigma^2_x$ or $\sigma^2_{\text{MC-dropout}}$ alone would result in an over-estimation of the error.

MC-dropout can be seen as a proxy to perform stochastic forward pass through the RNN models, where each weight row is dropped with probability determined by the magnitude of the data. Gaussian noise was added to data which were not dropped. The noise has a Gaussian distribution with zero mean and the variance was reckoned via MLE. RNNs were used to train the noise data to minimize the likelihood loss. It was also assumed that the mean and variance of the data uncertainty depended on climate data as the input variables. The estimated mean was equal to the mean of residuals between the observed data and the regression value of data. The variance indicated how much the residual between the data and the regression value fluctuates above or below the mean over time. High noise in simulation might reflect the fact that the data might obey other single probabilistic distributions or mixed distribution of multiple single distributions.

To validate the accuracy of simulations, P-factor, R-factor, and TUI (see supporting information) were used to compare the results. The width of the confidence interval is one of the most commonly used evaluation indices for addressing model uncertainty. The lower the width of the interval, the

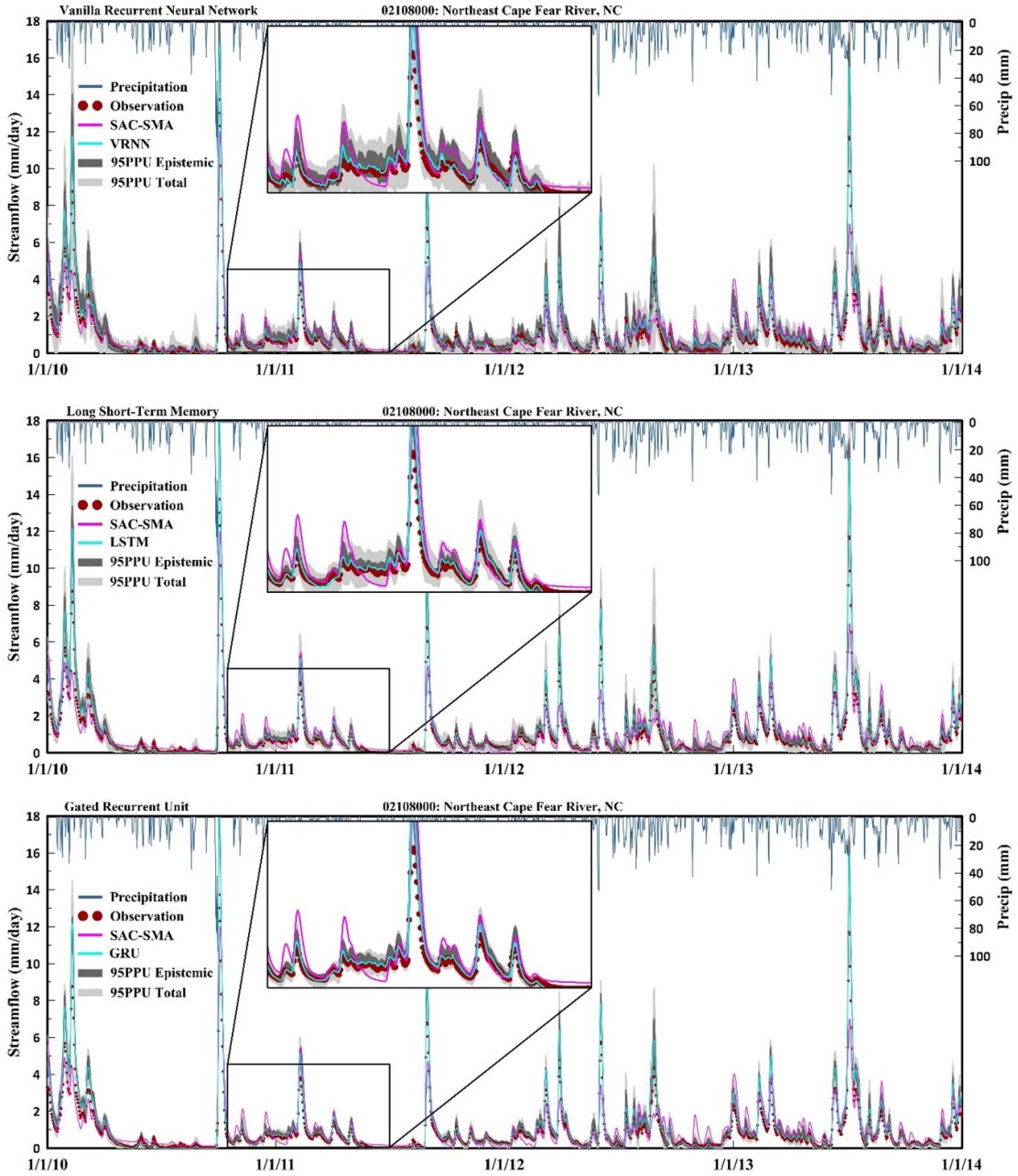


**Figure 7.** 95PPU ranges of simulated daily streamflow using the RNNs and SAC-SMA. Red dots, blue and dark shaded area denote observed data, epistemic 95PPU band, and 95PPU for the combination of epistemic and aleatoric uncertainties $\left(\sigma^2_{\text{comb}} = \sigma^2_x + \sigma^2_{\text{MC-dropout}}\right)$, respectively. No uncertainty assessment has been implemented for the SAC-SMA.

better the simulation effect can be obtained. Our experimental results showed that the total 95% confidence interval $(\sigma^2_x + \sigma^2_{\text{MC-dropout}})$ can appropriately bracket observed streamflow records in all three algorithms (figure 7). A comparison among different models revealed that the GRU narrowed the 95% uncertainty interval indicating a higher performance and lower uncertainty and errors in simulation. The blue shaded area also illustrates the 95% predictive uncertainty (95PPU) of the epistemic uncertainty, while the aleatoric term could be computed by differentiating the total and epistemic uncertainties $(\sigma^2_x = \sigma^2_{\text{comb}} - \sigma^2_{\text{MC-dropout}})$.

As shown in figure 7, the streamflow simulations follow a recurrent pattern for almost every water year. This allowed the RNNs to learn the data patterns and performed accurately. The 95PPU intervals bracketed most of the observed flows in each RNN model. Summary results and the performances of different RNNs employed in this research with respect to the total uncertainty and each individual term are provided in table 1. As shown, 95PPU band for the vanilla RNN and LSTM models bracketed >97% of observed data (P-factor) while GRU captured 84% of observations. In addition, GRU provided better estimation for the average width of the total and each individual

**Table 1.** Uncertainty performance metrics for the RNNs and the SAC-SMA with respect to epistemic and aleatoric uncertainties and their combination.

| | Vanilla RNN | | | LSTM | | | GRU | | |
|---|---|---|---|---|---|---|---|---|---|
| Uncertainty source | P-factor | R-factor | TUI | P-factor | R-factor | TUI | P-factor | R-factor | TUI |
| Epistemic | 0.77 | 0.51 | 1.52 | 0.55 | 0.28 | 1.96 | 0.52 | 0.26 | 2.00 |
| Aleatoric | 0.95 | 0.70 | 1.36 | 0.87 | 0.54 | 1.60 | 0.67 | 0.30 | 2.22 |
| Total | 0.98 | 1.14 | 0.86 | 0.97 | 0.80 | 1.22 | 0.84 | 0.55 | 1.53 |

term of uncertainty band (R-Factor). With respect to each individual term of uncertainty, we found that the aleatoric ($\sigma_x^2$) was slightly larger than epistemic ($\sigma^2_{\text{MC-dropout}}$) uncertainty in all simulations.

The TUI values of each individual uncertainty terms and their combination showed that GRU simulated streamflow with less uncertainty compared to the other two. In addition, GRU required less iterations than other networks and was able to model rainfall–runoff dynamical processes appropriately. This network was capable of predicting daily streamflow data from meteorological observations with accuracy comparable to a well-established SAC-SMA hydrologic model. GRU also converged quicker, and its recurrent networks were relatively faster than others. This is mainly because the GRU model directly uses all hidden states without control, and presents fewer computational parameters compared to the LSTM and vanilla RNN. Furthermore, GRU uses the hidden state to transfer information and it is capable of learning both short- and long-term dependencies with two gates, a reset gate and update gate. The reset gate helps GRU to capture short-term dependencies in time series data while the update gate can capture long-term dependencies in time series data. It is interesting to note that, the optimization results obtained by parameter adjustment for different rainfall–runoff data and learning networks can be different. Therefore, care should be exercised in using the results obtained from the GRU. Regarding the accuracy, the GRU showed fewer negative outliers and thus seems to be more robust network.

Focusing on the flow duration curve (FDCs) (see figure 8), all RNN networks precisely simulated streamflow fluctuations (high and low flows). FDC analysis further revealed that both high- and low-flow segment estimations were skillfully captured by both GRU and LSTM. High- and low- flow signatures in the streamflow hydrograph reflect respectively, as 'fast' and 'slow' runoff response/processes associated with impervious area runoff, surface runoff, interflow, and baseflow (see Yilmaz *et al* 2008 for further information). However, the flow exceedance probability curves associated with all three RNNs seem to be sensitive to extremely low flow events. The midsegment FDCs for all three RNNs as well as the SAC-SMA model do not seem to have a steep slope which reflect a catchment system with moderate to slow behavior and more sustain rainfall–runoff response. Overall, FDCs commonly used to indicate and classify watershed functioning. It also summarizes the catchment's ability to produce runoff values of different magnitudes. FDCs seem to be relatively insensitive to moderate and somewhat high flow events while very sensitive to low flow fluctuations (baseflow contribution).

The low-flow segment of the FDC contains information related to long-term sustainability of streamflow that is controlled by the interaction of baseflow with riparian ET during extended dry periods. Riparian area of the study catchment is covered by extensive alluvial and nonalluvial forested wetlands and wide floodplains. This causes substantial fluctuations in low flow values in which RNNs and SCA-SMA simulations were tempered by this heterogeneity. Since RNNs (as well as SAC-SMA) do not deal with physical processes (soil moisture, evapotranspiration, complexities in storage capacity, etc) of the catchment system, it seems computing low flow events when shallow aquifer is the primary contributor to river discharge (see Samadi *et al* 2017), is especially challenging for both RNNs and SAC-SMA models.

Like SAC-SMA, RNN networks have no exponential outflow function and thus the simulation value can be easily dropped to minuscule numbers or even zero. Our strategy for improving low flow simulations was to introduce an additional parameter and limit the simulated streamflow by greater than zero values, but to the minimum observed flow. This simple solution led to better FLV values for all RNNs used in this research, although other metrics such as NSE, were practically unaltered. Performance results of all the models are presented in table 2.

Regarding the SAC-SMA simulation (see figure 8), it seems the model was capable of simulating intermediate streamflow while relatively underestimated and overestimated low flow and high flow events, respectively. SAC-SMA uses a lumped fashion to calibrate soil moisture states and the basin's relative permeability. These states introduce as several key basin's factors into the model such as interflow, evapotranspiration, and percolation. It seems hydrologic variables such as loss (due to dense vegetation) and soil moisture have a strong control on rainfall–runoff processes and mechanism in the study region. These factors determine catchment wetness conditions and ET process. Thus, runoff magnitude might

**Table 2.** Accuracy metrics of the SAC-SMA and RNN models.

| | Model | NSE | KGE | TRMSE | ROCE | FHV | FMS | FLV |
|---|---|---|---|---|---|---|---|---|
| Calibration | SAC-SMA | **0.79** | 0.82 | **0.29** | 0.027 | −13.14 | **−10.68** | 27.19 |
| | Vanilla RNN | 0.71 | 0.83 | 0.32 | 0.03 | 1.30 | −13.77 | **−3.64** |
| | LSTM | 0.73 | 0.83 | 0.32 | **0.005** | **0.45** | −24.45 | −30.45 |
| | GRU | **0.79** | **0.86** | **0.29** | 0.019 | −4.64 | −28.26 | −14.77 |
| Validation | SAC-SMA | 0.77 | 0.79 | 0.28 | 0.112 | −17.88 | −0.89 | 39.19 |
| | Vanilla RNN | 0.68 | 0.84 | 0.3 | 0.01 | 4.00 | −1.49 | −6.72 |
| | LSTM | 0.77 | 0.87 | 0.28 | 0.034 | 3.18 | −18.95 | −16.32 |
| | GRU | **0.8** | **0.89** | **0.25** | **0.003** | **−0.43** | −14.8 | **−3.2** |

*Note*: Bold values indicate the best performance.

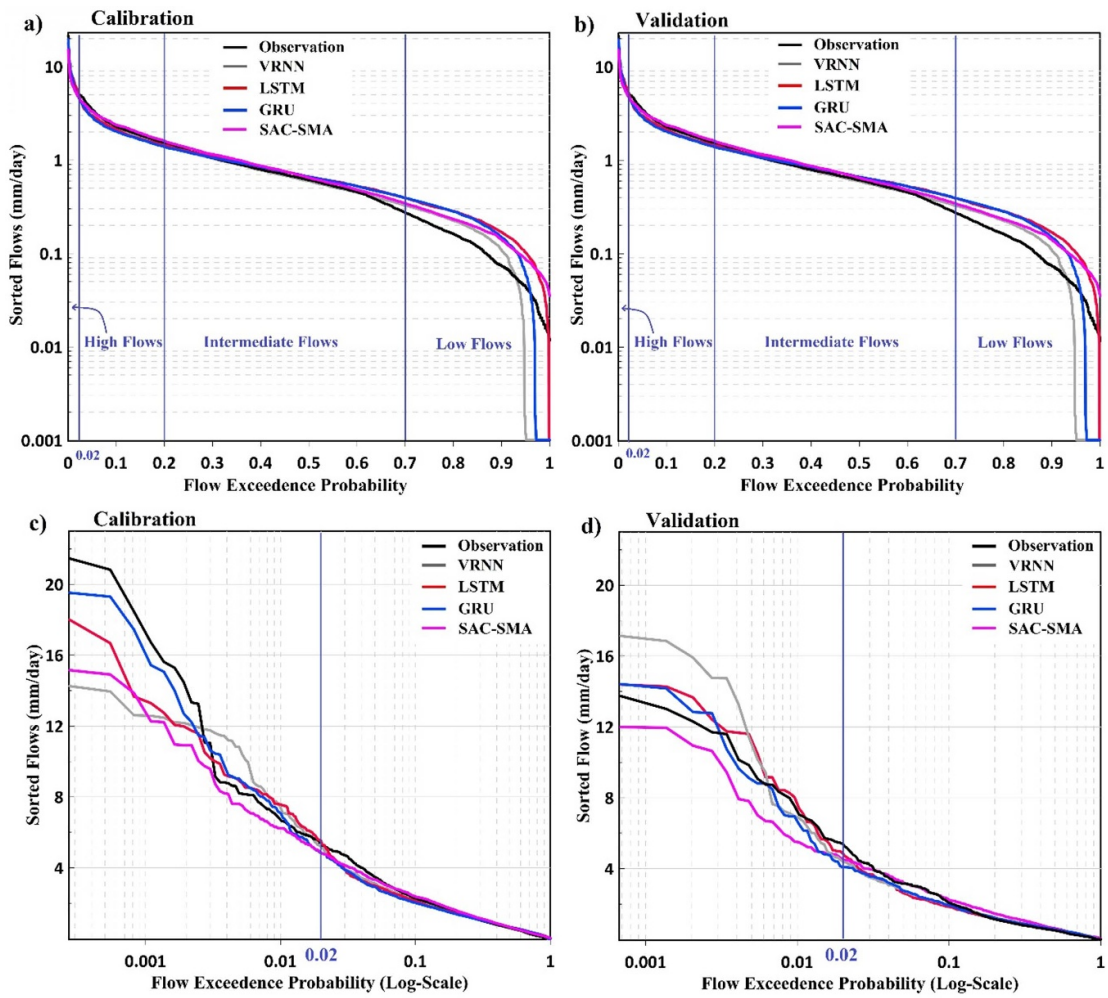


**Figure 8.** FDCs as diagnostic metrics to assess the performance of daily streamflow simulation modelled by the RNNs and SAC-SMA.

abruptly increase when catchment initial abstraction threshold was exceeded. In addition, catchment response time to rainfall showed significant variability in different seasons and years. It is interesting to note that in the coastal plain catchments a shallow aquifer system contributes continuously to the river system particularly during dry periods (see Samadi *et al* 2017). These contributions make surface and subsurface interactions and parametrization, as well as simulation mechanism itself a challenging task as discussed in Amatya and Jha (2011) and Samadi *et al* (2018).

## 4. Conclusion

This study examined multiple RNN algorithms for daily streamflow simulation and discussed how epistemic and heteroscedastic uncertainty can be quantified using a variational Bayesian approximation. Despite the challenges associated with a complex coastal

plain simulation, both GRU and LSTM appear to be the best tools to obtain comparable performances with the SAC-SMA model. Both algorithms provided the potential ability to simulate rainfall–runoff processes. Although, controversy regarding the use of these recurrent networks persists, we were able to improve the modeling performances in terms of data and model structure uncertainty. Nevertheless, we were less successful at improving low flow simulation (FDC low segment). There is some evidence that this challenge may be related to inherent fluctuations in the low flow data particularly with regard to riparian storage mechanism and shallow aquifer contribution to the river system (see Samadi *et al* 2017, 2018). This can be also interpreted as proof that the streamflow prediction error is stronger in some portions of timeseries where the networks may behave chaotically due to more ill-behaved data. The data noise uncertainty estimation is accustomed by data due to RNN data-driven nature. Thus, it makes data noise uncertainty estimation vulnerable to bias during training period. This observation shows an intrinsic limitation to any purely data-driven approach that data patterns and fluctuations may not be able to reveal rainfall–runoff variability. This obstacle could potentially be diminished by the future integration of the knowledge process-based models with an appropriate observation noise distribution. For example, process-based models could be constructed to determine several features based on physical relationships/attributes (runoff generation mechanism, flow resistance factors, travel time, etc) that were not adequately represented in the training data. How to appropriately couple physical attributes with different classes/structures of RNNs is still an open question (Karpatne *et al* 2017, Shen *et al* 2018). In addition, other approaches such as Stein variational gradient descent training (Liu and Wang 2016, Mo *et al* 2018) can be included to the RNN structure as a powerful approximate inference algorithm by making parametric assumptions about the form of data distribution. This can enhance the performance of observation uncertainty estimation, as well as the distinction between network weight uncertainty data noise.

We also performed an intercomparison analysis with continental scale RNN studies to evaluate the modeling performances developed in our site-specific study. In terms of simulation performances, both LSTM and GRU models employed in this research more accurately simulated the streamflow values (NSE > 0.77) than over 50% of the catchments/case studies investigated by (Kratzert *et al* 2019, Feng *et al* 2020, Jiang *et al* 2020). In our case, both GRU and LSTM algorithms provided the potential ability to simulate a complex yet nonlinear rainfall–runoff process across a coastal plain drainage system. Focusing on a local catchment, we were able to improve the model performance in terms of data uncertainty and model structure uncertainty, although we were less successful at improving LSTM and GRU simulations for low flow modeling (FDC low segment). In this region, shallow water tables may severely restrict the amount of water recharging to the river system during low flow events. Shallow aquifers along with poor natural drainage created an excessive soil water condition that is difficult for recurrent networks to understand and capture their dynamics and interaction. The quantity of water in the shallow aquifer system affects the variable source area involved in the generation of saturation overland flow and this mechanism can cause the recurrent networks to behave chaotically during low flow events.

Another difficulty that concerned us was the generalization of both GRU and LSTM predictive uncertainty to domain shift where time series data come with missing patterns. In this case, it is important to measure 'if the network knows what it knows'. For instance, if a network trained on a set of streamflow data but is evaluated on a completely different dataset, then the network should output high predictive uncertainty as inputs from a different dataset that would have different pattern than the training data. However, the choice of the likelihood function and the probability distribution of the data should perceive further attention in the future. Often, the assumption of a Gaussian distribution is valid in a catchment simulation problem where the rainfall–runoff processes follow a linear approach. For nonlinear rainfall–runoff simulation, the limitation of Gaussian distribution assumption for the likelihood function could be potentially enhanced with a proper noise distribution. We acknowledge that both LSTM and GRU simulations can be made more efficient by using more robust Bayesian inference such as Bayesian Model Averaging (BMA) with fixed and flexible prior distributions (see Samadi *et al* 2020) and/or Markov Chain Monte-Carlo optimization methods (Duane *et al* 1987) addressing both aleatoric and epistemic uncertainties. As always, we invite dialogue with geoscience, engineering, and data science communities interested in related DL simulation problems.

## Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://zenodo.org/record/5534879#.YW1oxxrMJPY.

## Acknowledgments

This work was supported by the U.S. National Science Foundation (NSF) Directorate for Engineering under Grant No. CBET 1901646. The data that supports the findings of this study is openly available at zenodo (https://zenodo.org/record/5534879#.YW1r9BrMJPY).

## ORCID iDs

S Sadeghi Tabas https://orcid.org/0000-0001-9157-3397
S Samadi https://orcid.org/0000-0003-1494-6481

## References

Addor N, Newman A J, Mizukami N and Clark M P 2017 The CAMELS data set: catchment attributes and meteorology for large-sample studies *Hydrol. Earth Syst. Sci.* **21** 5293–313

Amatya K M and Jha M K 2011 Evaluating the SWAT model for a low-gradient forested watershed in Coastal South Carolina *Trans. Am. Soc. Agric. Biol. Eng.* **54** 2151–63

Bengio Y 2012 Practical recommendations for gradient-based training of deep architectures *Neural Networks: Tricks of the Trade* (Berlin: Springer) pp 437–78

Blundell C, Cornebise J, Kavukcuoglu K and Wierstra D 2015 Weight uncertainty in neural network. *Int. Conf. on Machine Learning* (*June*) (PMLR) pp 1613–22

Casdagil M 1989 Nonlinear prediction of chaotic time series *Physica* D **35** 335–56

Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Learning phrase representations using RNN encoder-decoder for statistical machine translation (arXiv:1406.1078)

Damianou A and Lawrence N 2013 Deep Gaussian processes *Proc. 16th Int. Conf. on Artificial Intelligence and Statistics* pp 207–15

der Kiureghian A and Ditlevsen O 2009 Aleatory or epistemic? Does it matter? *Struct. Saf.* **31** 105–12

Duane S, Kennedy Anthony D, Pendleton Brian J and Roweth D 1987 Hybrid Monte Carlo *Phys. Lett.* B **195** 216–22

Fang K, Kifer D, Lawson K and Shen C 2020 Evaluating the potential and challenges of an uncertainty quantification method for long short-term memory models for soil moisture predictions *Water Resour. Res.* **56** e2020WR028095

Feng D, Fang K and Shen C 2020 Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales *Water Resour. Res.* **56** e2019WR026793

Feng D, Lawson K and Shen C 2021 Mitigating prediction error of deep learning streamflow models in large data-sparse regions with ensemble modeling and soft data *Geophys. Res. Lett.* **48** e2021GL092999

Gal Y and Ghahramani Z 2016a Dropout as a Bayesian approximation: representing model uncertainty in deep learning *Int. Conf. on Machine Learning* pp 1050–9

Gal Y and Ghahramani Z 2016b Bayesian convolutional neural networks with Bernoulli approximate variational inference *ICLR Workshop Track* (arXiv:1506.02158)

Graves A 2011 Practical variational inference for neural networks *Advances in Neural Information Processing Systems* pp 2348–56

Haykin S and Principe J 1998 Making sense of a complex world *IEEE Signal Process. Mag.* **15** 66–81

Hernandez-Lobato J, Li Y, Rowland M, Bui T, Hernández-Lobato D and Turner R 2016 Black-box alpha divergence minimization *Int. Conf. on Machine Learning* (*June*) (PMLR) pp 1511–20

Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

Jiang S, Zheng Y and Solomatine D 2020 Improving AI system awareness of geoscience knowledge: symbiotic integration of physical approaches and deep learning *Geophys. Res. Lett.* **47** e2020GL088229

Jordan M I, Ghahramani Z, Jaakkola T S and Saul L K 1999 An introduction to variational methods for graphical models *Mach. Learn.* **37** 183–233

Karpatne A, Atluri G, Faghmous J H, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N and Kumar V 2017 Theory-guided data science: a new paradigm for scientific discovery from data *IEEE Trans. Knowl. Data Eng.* **29** 2318–31

Kendall A and Gal Y 2017 What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems* **30** 5575–85 (available at: https://arxiv.org/abs/1703.04977v2)

Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

Kratzert F, Klotz D, Brenner C, Schulz K and Herrnegger M 2018 Rainfall–runoff modelling using long short-term memory (LSTM) networks *Hydrol. Earth Syst. Sci.* **22** 6005–22

Kratzert F, Klotz D, Shalev G, Klambauer G, Hochreiter S and Nearing G 2019 Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets *Hydrol. Earth Syst. Sci.* **23** 5089–110

LeCun Y A, Bottou L, Orr G B and Müller K-R 2012 *Efficient Backprop, Neural Networks: Tricks of the Trade* (Berlin: Springer) pp 9–48

Liu Q and Wang D 2016 Stein variational gradient descent: a general purpose Bayesian inference algorithm *Advances In Neural Information Processing Systems* (Barcelona, ES) pp 2378–86

Minns A and Hall M 1996 Artificial neural networks as rainfall-runoff models *Hydrol. Sci. J.* **41** 399–417

Mirikitani D T and Nikolaev N 2010 Recursive Bayesian recurrent neural networks for time-series modeling *IEEE Trans. Neural Netw.* **21** 262–74

Mo S, Zhu Y, Zabaras N, Shi X and Wu J 2018 Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media *Water Resour. Res.* **55** 703–28

Rahmani F, Lawson K, Ouyang W, Appling A, Oliver S and Shen C 2021 Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data *Environ. Res. Lett.* **16** 024025

Rasmussen C E and Williams C K I 2006 *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* vol 2 (Cambridge, MA: MIT Press) p 4

Samadi S, Pourreza-Bilondi M, Wilson C A and Hitchcock D B 2020 Bayesian model averaging with fixed and flexible priors: theory, concepts, and calibration experiments for rainfall-runoff modeling *J. Adv. Model. Earth Syst.* **12** e2019MS001924

Samadi S, Tufford D and Carbone G 2017 Assessing prediction uncertainty of a semi-distributed hydrology model for a shallow aquifer dominated environmental system *J. Am. Water Res. Assoc.* **53** 1368–89

Samadi S, Tufford D and Carbone G 2018 Estimating hydrologic model uncertainty in the presence of complex residual error structures *Stochastic Environ. Res. Risk Assess.* **32** 1259–128

Shen C 2018 A transdisciplinary review of deep learning research and its relevance for water resources scientists *Water Resour. Res.* **54** 8558–93

Werbos P J 1990 Backpropagation through time: what it does and how to do it *Proc. IEEE* **78** 1550–60

Yilmaz K K, Gupta H V and Wagener T 2008 A process-based diagnostic approach to model evaluation: application to the NWS distributed hydrologic model *Water Resour. Res.* **44**

Zhu S, Xu Z, Luo X, Liu X, Wang R, Zhang M and Huo Z 2021 Internal and external coupling of Gaussian mixture model and deep recurrent network for probabilistic drought forecasting *Int. J. Environ. Sci. Technol.* **18** 1221–36