Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

Using a physics-based hydrological model and storm transposition to investigate machine-learning algorithms for streamflow prediction

Faruk Gurbuz[a,b,*], Avinash Mudireddy[c], Ricardo Mantilla[d], Shaoping Xiao[e]

[a] *General Directorate of State Hydraulic Works, Ankara, 06530, Turkiye*
[b] *IIHR-Hydroscience & Engineering, University of Iowa, Iowa City, IA 52242, USA*
[c] *The Iowa Initiative for Artificial Intelligence, University of Iowa, Iowa City, IA 52242, USA*
[d] *Department of Civil Engineering, University of Manitoba, Winnipeg, MB R3T2N2, Canada*
[e] *Department of Mechanical Engineering, Iowa Institute of Technology, University of Iowa, Iowa City, IA 52242, USA*

ARTICLE INFO

This manuscript was handled by Nandita Basu, Editor-in-Chief, with the assistance of Juliane Mai, Associate Editor

Keywords:
Streamflow prediction
Artificial Intelligence
Scale independent performance metrics

ABSTRACT

Machine learning (ML) algorithms have produced remarkable advances in streamflow prediction, exceeding the performance of calibrated conceptual and physics-based hydrological models that have been developed over many decades. ML algorithms seem to overcome the issue of errors known to be present in rainfall and streamflow estimates that have hindered the performance of hydrological models for decades. In this paper, we propose a methodology for testing and benchmarking ML algorithms using artificial data generated by physically-based hydrological models. Our approach makes it possible to design controlled numerical experiments that can improve our understanding of this new generation of black-box models. We conducted a diagnostics study to demonstrate our methodology in which we attempted to determine if ML algorithms can identify a function relating streamflow and rainfall. This exercise combined the implementation of the distributed hillslope-link hydrological model (HLM) on a 4,385 km^2 basin driven by precipitation fields created using the stochastic storm transposition (SST) framework, and an advanced deep learning algorithm based on gated recurrent unit (GRU)-Attention neural networks. The data generated allowed us to create prediction scenarios that are equivalent to the hindcast and real-time forecast problems. We proposed a set of scale-independent performance metrics to evaluate the results of our experiment and found that the GRU can correctly identify a predictive function for all analyzed locations in the river network. We concluded that under the circumstances tested in this study, deep learning can identify the transformation function when trained in Hindcast Mode, making it a powerful tool to determine the streamflow response of a basin to predetermined rainfall scenarios. However, it fails to significantly outperform the predictions of temporal persistence when tested in Forecast Mode.

1. Introduction

Rapid advancements in machine learning (ML) and deep learning (DL) are impacting every field of science and technology, including hydrologic forecasting of streamflow. Recurrent neural networks (RNNs) (Haykin, 1999) are designed to predict time series that depend on input forcings over a historical period and have shown potential in predicting streamflow. Existing studies of ML/DL techniques have reported performance metrics that traditional approaches of conceptual or physically based hydrologic modeling cannot reliably achieve (Mai et al., 2022). This performance is promising across different types of basins and sidesteps issues that have affected hydrological modeling for many years, namely the fact that rainfall observations used to force hydrological models are estimates subject to spatial and temporal errors and that streamflow time series used to calibrate and validate models are themselves estimates based on rating curves that are also subject to error. Therefore, ML algorithms can identify the relationship between inputs and outputs in a river basin while at the same time ascertaining appropriate corrections for the errors present in rainfall and streamflow time series. However, it is difficult to extract meaningful interpretations of the relationships identified by DL training because RNNs are essentially black boxes.

Although recent success in streamflow prediction has been achieved with RNNs, there is a long history of applying ML algorithms to

[*] Corresponding author at: General Directorate of State Hydraulic Works, Ankara, 06530, Turkiye.
E-mail address: farukgurbuz@dsi.gov.tr (F. Gurbuz).

https://doi.org/10.1016/j.jhydrol.2023.130504
Received 7 May 2023; Received in revised form 29 October 2023; Accepted 7 November 2023
Available online 22 November 2023

streamflow forecasting. Early works include linear or non-linear regression models for runoff prediction (Granata et al., 2016) and artificial neural networks (ANNs) to model hydrological systems (Oyebode and Stretch, 2019). Estévez et al. (2020) used wavelet neural networks, which combined neural networks with the theory of wavelets to predict monthly precipitation for water resources management. Furthermore, Vidyarthi et al. (2020) investigated the effects of two popular ANN training algorithms, the gradient descent (GD) and Levenberg-Marquardt (LM) algorithms, on weighting parameters of neural networks when approximating the rainfall-runoff process. In that work, they proposed a novel method consisting of basic statistics to assess the sensitivity of ANN parameters. RNNs, by comparison, consider the current input and what it has learned from the previously supplied input data through embedded memory. Kumar et al. (2004) tested both ANN and RNN to forecast the monthly flows of a river in India. They concluded that the RNN gave better results for both single-step ahead and multiple-step ahead forecasting. In addition, Wan et al. (2019) implemented an Elman neural network (a type of RNN) in a real-time framework for probabilistic flood forecasting. They demonstrated that the proposed method is highly practical for providing decision-making support in flood control. However, a naïve RNN suffers from issues of exploding and vanishing gradients. The exploding gradient issue occurs when gradients during neural network training become excessively large, leading to unstable learning and divergent behavior. Conversely, the issue of vanishing gradient arises when gradients become extremely small, impeding the training process by slowing down weight updates and hindering convergence.

An improved RNN, called long-short term memory (LSTM), was designed to resolve these issues (Hochreiter and Schmidhuber, 1997) by learning long-term dependencies between the input and output of the network. Kratzert et al. (2018) specifically investigated the potential of LSTM for hydrological modeling applications. They found that LSTM has a better performance compared to a lumped hydrological model, namely the Sacramento Soil Moisture Accounting Model (SAC-SMA), coupled with the Snow-17 snow model. Hu et al. (2018) proposed an LSTM-based data-driven approach for flood forecasting. However, neither of these works predicted multiple-step and continuous output targets. To solve this issue, Sutskever et al. (2014) proposed a sequence-to-sequence (seq2seq) learning architecture, also called encoder-decoder LSTM networks. Building on this work, Xiang et al. (2020) developed an LSTM-seq2seq model, which could continuously predict runoff for 24 h. The advantage of an LSTM is that it remembers both long-term and short-term patterns in the data. Cho et al. (2014) proposed a gated recurrent unit (GRU), which maintains the advantages of LSTM but with fewer gates, meaning they can be trained more quickly. In this work, they developed and examined a new neural network architecture using GRU cells with attention mechanisms (Bahdanau et al., 2014) to predict streamflows. It should be noted that the developed neural network uses the same encoder-decoder model as in Xiang et al. (2020). Specifically, the encoder encapsulates the information of all input elements into a fixed-length context vector, while the decoder learns the context from the previous cell in the sequence. In the present study, we use GRUs instead of LSTMs, and our proposed architecture employs the attention mechanism (Bahdanau et al., 2014). This approach provides a richer context so the decoder can learn where to pay attention to the context vector in a weighted manner. In addition, the runoffs at multiple locations on the river network are predicted hierarchically from upstream to downstream.

Most studies on ML applications have used real data for training the algorithms. The generalization capability of these models hinges upon the diversity and extent of the available datasets, encompassing various regions, time scales, and basin scales. In this paper, we designed a controlled experiment to test the ability of ML algorithms to predict flows under various circumstances, including different basin scales, differences in upstream availability of information, and variations in the quality of forecasted future rainfall. To this end, we implemented the distributed hillslope-link model (HLM) (Mantilla et al., 2022) in the Turkey River basin (4,835 km^2) to create a *virtual environment* and generate a synthetic data set of precipitation and streamflow time series. This strategy was previously used in a study by Perez et al. (2019) to investigate properties of peak flow distributions across basin scales. The hydrological model captures two key aspects of the physics of runoff generation and transport. The first is the non-linear relationship between precipitation intensity and antecedent conditions to runoff generation, and the second is the movement of water along the complex river network that drains the landscape. The Turkey River Basin has been discretized into over 10,000 hillslope scale control volumes interconnected by an equal number of channel links. The equations that control the rainfall-runoff partitioning have been previously reported in the literature, along with an evaluation of the model's ability to reproduce actual observations (Fonley et al., 2021; Velásquez et al., 2021). Our goal is to test the ability of ML algorithms to predict streamflow time series generated by a distributed hydrological model given mean-areal rainfall and upstream information for different locations in the river network. The controlled numerical experiment reported in this paper allows us to (*i*) develop fair performance metrics that can be compared across scales, (*ii*) determine the value of different inputs, and (*iii*) quantify the expected difference in the ML performance in hindcast and forecast modes. We aim to make our controlled experiment simple enough to be tractable but with a complexity that requires DL.

2. Virtual basin setup

2.1. Turkey river basin

This study employs a virtual representation of a river network that has been configured to correspond with the Turkey River Basin, which is a location we have studied previously (Mantillla et al., 2021; Perez et al., 2019; Wright et al., 2017). The basin is in northeastern Iowa in the midwestern United States (Fig. 1) and drains into the Mississippi River bordering the State of Iowa on the east. It has a total drainage area of 4,385 km^2. There are originally five stream gauges operated by the United States Geological Survey (USGS) across the basin at the locations denoted by circles in Fig. 1. Therefore, we assume that these locations are the streamflow monitoring sites in our virtual environment. We only use the portion of the basin upstream of Garber as it is the monitoring location that is furthest downstream on the Turkey River. The basin consists of 10,642 hillslope-channel link units extracted using a 90-meter resolution Digital Elevation Model (DEM). The average hillslope area and channel-link length of the decomposed basin are 0.38 km^2 and 0.71 km, respectively. It should be noted that the same river connectivity is also employed in the streamflow forecasting scheme of the Iowa Flood Center (Krajewski et al., 2017).

2.2. Rainfall input (Storm Transposition)

We used the Stochastic Storm Transposition (SST) procedure (Wright et al., 2013) to generate spatiotemporally realistic rainfall events in our virtual environment. The SST framework aims to increase the length of extreme rainfall scenarios in a particular area by using the existing rainfall information from the neighboring regions, in which the characteristics of extreme rainfall events are homogenous (Wright et al., 2013). The use of SST in reconstructing long-term climatology of extreme rainfall events in a region of interest has received considerable attention due to the onset of high-resolution bias-corrected quantitative rainfall estimates by weather radars (Heiss et al., 1990; Krajewski and Smith, 2002). In addition to rainfall frequency analysis (England et al., 2014), the SST procedure coupled with physics-based rainfall-runoff models is useful for flood frequency analysis (England et al., 2014; Mantillla et al., 2021; Perez et al., 2019; Yu et al., 2019). We refer interested readers to (Wright et al., 2020) for a more comprehensive review of the SST procedure and its practical use cases.

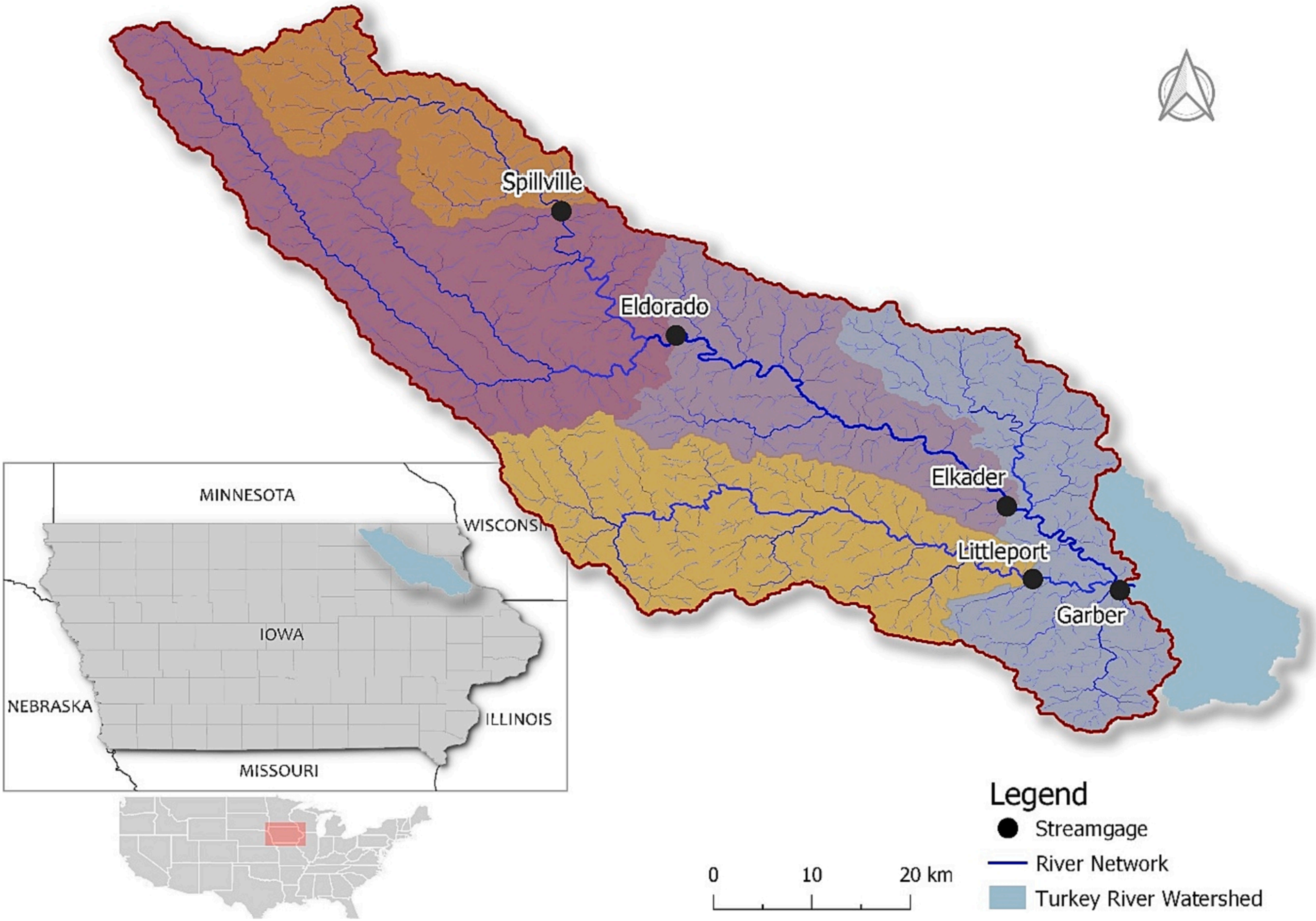

Fig. 1. The location of the Turkey River Basin, with sub-basins illustrated in different colors.

The rainfall data used in this study was generated using RainyDay, a Python-based open-source software that combines remotely sensed rainfall data with SST (Wright et al., 2017). The SST methodology of RainyDay is a five-step task: (1) Define a geographic transposition domain A' involving the basin of interest A. (2) Identify m number of temporally distinct storms in A' from n years of rainfall remote sensing data. It considers rainfall accumulation of duration t with the same size, shape, and orientation of A in order to establish so-called "storm catalogs". (3) Generate a random integer k, which corresponds to the number of storms for each year. The generation of k by RainyDay can be based on a Poisson distribution or an empirical distribution. (4) Randomly select k number of storms from the storm catalog. (5) Reiterate steps 3 and 4 based on a user-defined number (T_{max}) to produce T_{max} years of t-hour synthetic annual rainfall maxima for A.

We used gauge-corrected Stage IV rainfall data from 2002 to 2018 (April-November) to establish a storm catalog for temporal and spatial resampling. The transposition domain covered 94.25–89.25°W and 40.5–45.5°N, which aligns with those utilized in previous studies (Mantillla et al., 2021; Perez et al., 2019; Yu et al., 2019; Zhu et al., 2018). The storm catalog consisted of 320 of the most intense rainfall events within the domain. The selection of "most intense" events was based on 72-hours of rainfall accumulation over areas that have the same shape and size as the Turkey River Basin. The random integer k follows a Poisson Distribution with rate parameter λ, which is calculated by dividing the total number of events in the storm catalog by the number of years in the rainfall remote sensing data. The same remote sensing dataset and the RainyDay parameters have been previously used by (Mantillla et al., 2021) for SST-based regional flood frequency analysis. Therefore, the generated rainfall events have the same temporal and spatial resolution as Stage IV data (1 h, 4 km).

One of the objectives of this study is to determine the sources of information that are most valuable for the DL algorithm for streamflow prediction as well as the prediction skill of the proposed GRU models. Therefore, the realism of the spatiotemporal structure of rainfall events and the complexity of the hydrologic model in terms of rainfall-runoff transformation is significant.

2.3. Hydrologic model

We used the Hillslope-Link Model (HLM) to generate the streamflow time series in the virtual basin. The Iowa Flood Center (IFC) has employed the HLM as the main component of its statewide real-time flood forecasting system, which provides streamflow predictions for over 1,000 communities across the state (Krajewski et al., 2017; Mantilla et al., 2022). The HLM is a data-intensive, physics-based, parsimonious distributed hydrologic model that simulates rainfall-runoff processes at hillslopes (Krajewski et al., 2017; Mantilla, 2007; Mantilla et al., 2006; Quintero et al., 2020) and routes flow along the river network. The model uses a large set of ordinary differential equations to describe the physical processes, including initial abstraction, overland flow, infiltration, percolation, base flow, and streamflow routing (Quintero et al., 2020). In addition, the model configuration offers users a flexible environment for modifications in the source code and harnesses the benefits of parallel computing with its asynchronous solver that employs Runge-Kutta methods (Small et al., 2013).

The key component of the HLM is the decomposition of a drainage basin into pairs of hillslope and channel-link units (Mantilla and Gupta, 2005) (Fig. 2). The mass and momentum conservation equations are also applied at the hillslope scale. As illustrated in Fig. 2, the control volumes at each hillslope are expressed as the following storages: surface or ponded $s_p(t)[m]$, topsoil $s_t(t)[m]$, subsurface (i.e., soil) $s_s(t)[m]$ and channel-link $q(t)\left[m^3s^{-1}\right]$. In HLM, the effective rainfall or runoff coefficient is a non-linear function of the effective water depth in the topsoil storage. In addition, the overland flow from ponded storage on the hillslope surface q_{pc} and the baseflow from the hillslope subsurface storage q_{sc} contribute to streamflow. The water routing from upstream channel links to downstream locations is simulated using a power-law function in which water velocity is a function of upstream area and discharge. The underlying processes in HLM add sufficient spatial and temporal variability to the reproduced streamflow time series and therefore serve the objectives of this study.

2.4. Data collection

We forced the HLM with 5,000 synthetically generated but spatiotemporally realistic rainfall scenarios with an hourly temporal resolution to generate hourly streamflow time series in the virtual basin. We ran the HLM sequentially such that each rainfall event was fed into the hydrologic model successively. The model state at the end of one event becomes the initial condition for the next. Thus, we achieved a continuous streamflow time series. We dedicated a 20-day-long period for each rainfall-runoff event. The rainfall, however, occurs in the first 72 h of that time window. We found a length of 20 days is appropriate for a rainfall-runoff event because the topsoil becomes sufficiently drained during that period (see Fig. 3). As a result, the prior soil moisture conditions and the remaining control volume states have little to no effect on the subsequent event.

Depending on the spatial and temporal distribution of the rainfall events along with the antecedent soil moisture conditions, the resulting hydrographs present differences in terms of magnitude, duration, and volume as desired (see Fig. 3). Note that, as discussed in subsequent sections, we used basin-average rainfall time series from SSTs corresponding to the “gauged” basins as input to the proposed DL models. In other words, we processed spatially variable rainfall information to create a basin-average rainfall time series for the locations where the proposed DL models predict streamflows. The synthetic data produced by the HLM allows us to create two equivalent prediction scenarios: a hindcast and a real-time forecast problem. In the hindcast prediction scenario, both present and future rainfall are perfectly known. On the other hand, in the forecast problem, future rainfall is unknown and/or subject to uncertainty, which is also referred to as reforecast.

Based on the data collected, the inputs to the DL models are provided in the form of a three-dimensional time series sequence. The first dimension corresponds to the number of input samples, which are time-series data with a fixed timestep. The timestep is the frequency by which data is supplied to the models. In our case, the timestep is one hour, and there are a total of 2,400,000 data samples drawn from the results of 5,000 continuously-simulated events. The second dimension corresponds to the number of timesteps in each data sample. This number/size varies depending on the feature. If the features correspond to both past and future, then the size is $\tau_p + \tau_{f,}$ given τ_p is the number of past timesteps, and $\tau_{f,}$ is the number of future timesteps (i.e., the maximum lead time). On the other hand, if the feature corresponds to the past only, then the size is τ_p. We chose $\tau_p = 72$ h and $\tau_f = 24$ h in this study. The third dimension corresponds to the number of input features, including past and future precipitation of the current link, past and future streamflows of upstream sub-basins, and past streamflow of the current sub-basin (Throughout the study, the term “sub-basins” is used to denote the five selected “gauged” basins). Note that various models will be considered to study the role of different features (i.e., input information) in Section 3. Therefore, the size of this dimension may vary depending on the DL model studied.

3. Deep learning model description and training

3.1. Gru-based Encoder-Decoder attention network basics

In DL, GRUs are adapted versions of RNNs proposed by Cho et al. (2014) to address the problem of exploding/vanishing gradients in naïve RNNs. A GRU has a forget gate but not an output gate, so it has fewer parameters than LSTM. It has been demonstrated that GRUs perform similarly to LSTM on speech signal modeling and natural language processing (Ravanelli et al., 2018) but perform better in smaller and less

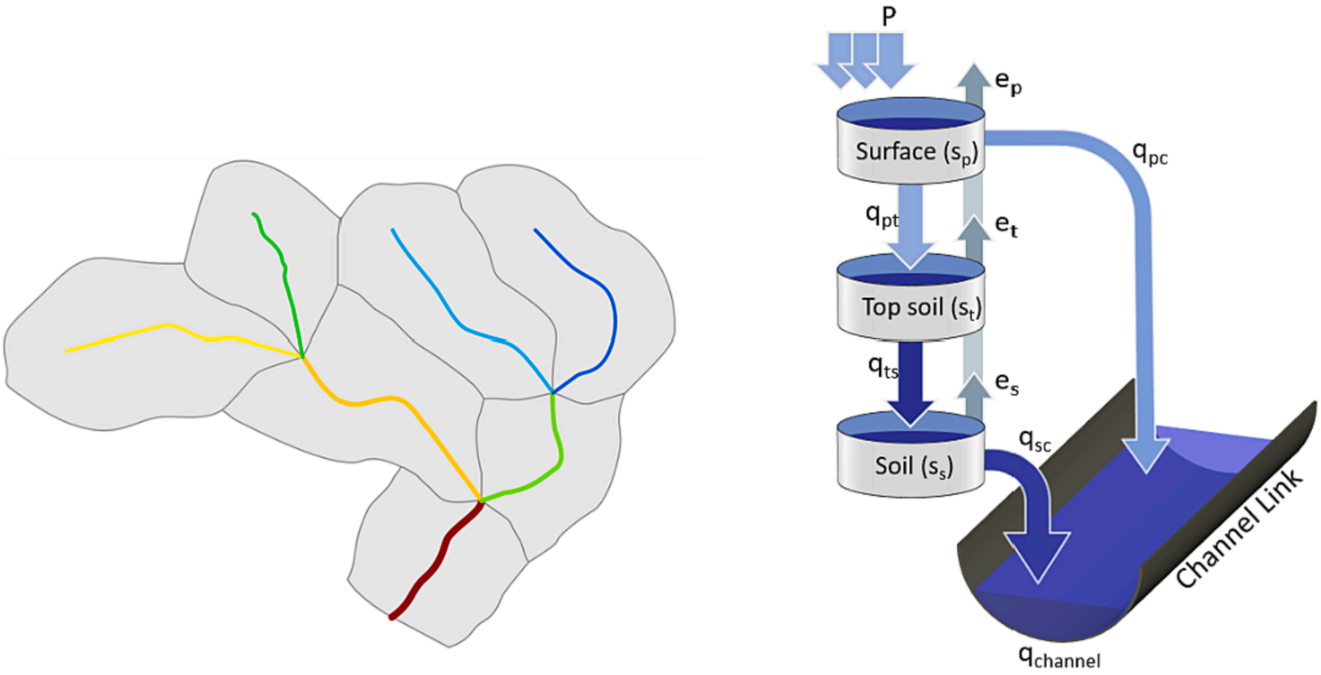

Fig. 2. The decomposition of a basin into hillslope-channel links (on the left) and the schematic of the Hillslope-Link Model (on the right).

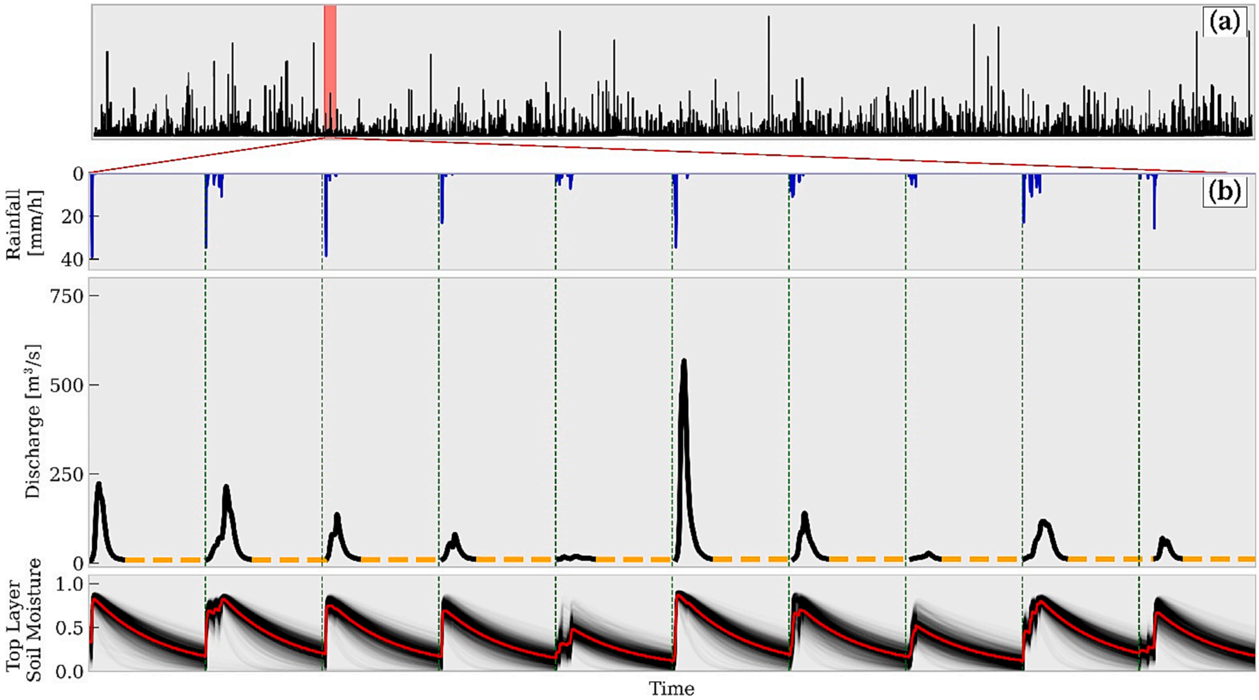

Fig. 3. An example of the hydrographs generated by coupling HLM and RainyDay. The upper panel (a) showcases 1,000 rainfall-runoff events. The lower panel (b) presents time series plots of basin-average rainfall, streamflow, and top layer soil moisture for 10 of these events at Spillville, marked by a red band in (a). Within (b), each rainfall-runoff event (20-day time window) is demarcated by a green dashed line. The streamflow's baseflow component is denoted by an orange dashed line, while a solid black line depicts direct runoff. The red solid line signifies the average soil moisture for the basin, while the remaining black lines in the background represent the soil moisture of the hillslopes upstream of Spillville. Note that 1 (0) corresponds to saturated (dry) soil. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

frequent datasets (Gruber and Jockisch, 2020). GRU networks preserve information and learn patterns in long sequences via their unique network design. Therefore, they are useful for many ML problems, including tasks like predicting gas concentrations using gas sensor readings as input (Wang et al., 2020). The utility of these models has also been demonstrated in the domain of streamflow prediction through recent studies (Ayzel and Heistermann, 2021; Gao et al., 2020; Ha et al., 2021; Muhammad et al., 2019).

A basic GRU unit or cell, as shown in Fig. 4, is composed of a reset gate and an update gate to control how the information flows in the different layers. Each cell constitutes

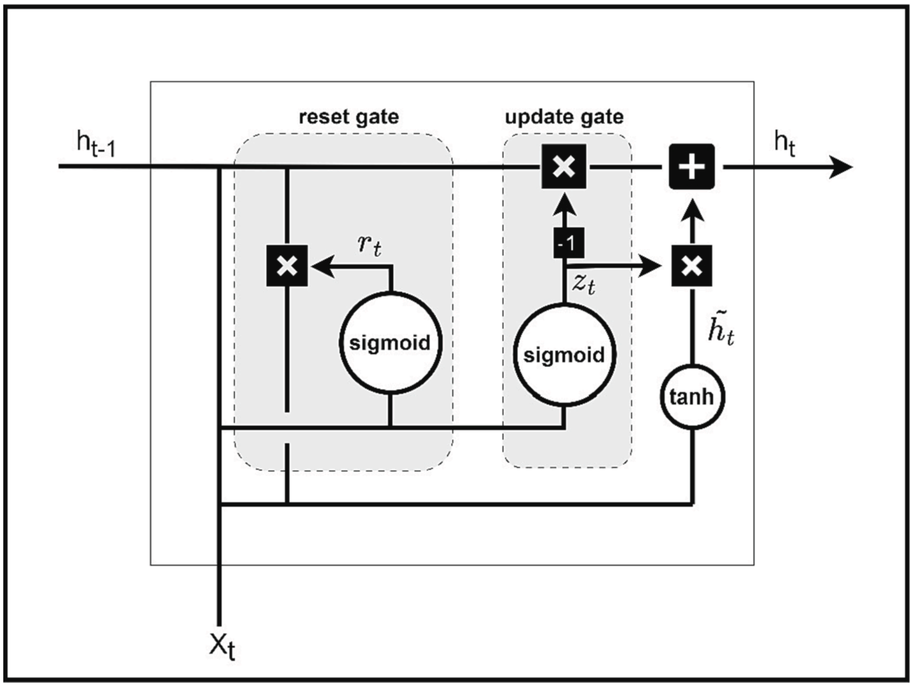

Fig. 4. A basic GRU unit.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

$$\widetilde{h}_t = tanh(W_h x_t + U_h(r_t \otimes h_{t-1}) + b_c\,) \tag{3}$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \widetilde{h}_t \tag{4}$$

where $x_t \in R^d$ is the input vector at time step t, and d refers to the number of input features. $z_t \in R^h$ is the update gate vector, and h is the number of hidden units. σ corresponds to the sigmoid function as the activation function and $\otimes$ is the Hadamard product operator. $r_t \in R^h$ is the reset gate vector. $\widetilde{h}_t \in R^h$ is the candidate activation vector, and $h_t \in R^h$ is the hidden/output state vector. $W \in R^{h\times d}$, $U \in R^{h\times h}$, and $b \in R^h$ are weight coefficient matrices and the bias vector, which need to be learned during neural network training.

In the proposed GRU encoder-decoder neural network (Cho et al., 2014; Sutskever et al., 2014), as shown in Fig. 5, the encoders encode the source time series to a fixed-length vector (hidden states), and the decoder maps the vector back to the target time series. Both the encoder and decoder layers contain GRU units as their cells. For example, given the input sequence $(x_1, x_2, ..., x_t)$ of length t, an encoder computes the corresponding sequence of hidden states $c = (h_1, h_2, ..., h_t)$, such that

$$h_t = f(x_t, h_{t-1}) \tag{5}$$

where f is a non-linear function composed of operations of Eqns (1)-(4) in unidirectional GRU cells. In our DL model for runoff prediction, x_t denotes the rainfall and streamflow observations at time t, h_t denotes the coded vector that contains all the necessary information from x_t, and c is the context vector.

The decoder is trained to predict the future runoff sequence y_{t+n} at time $t + n$, given the context vector c and all the previous runoff ob-

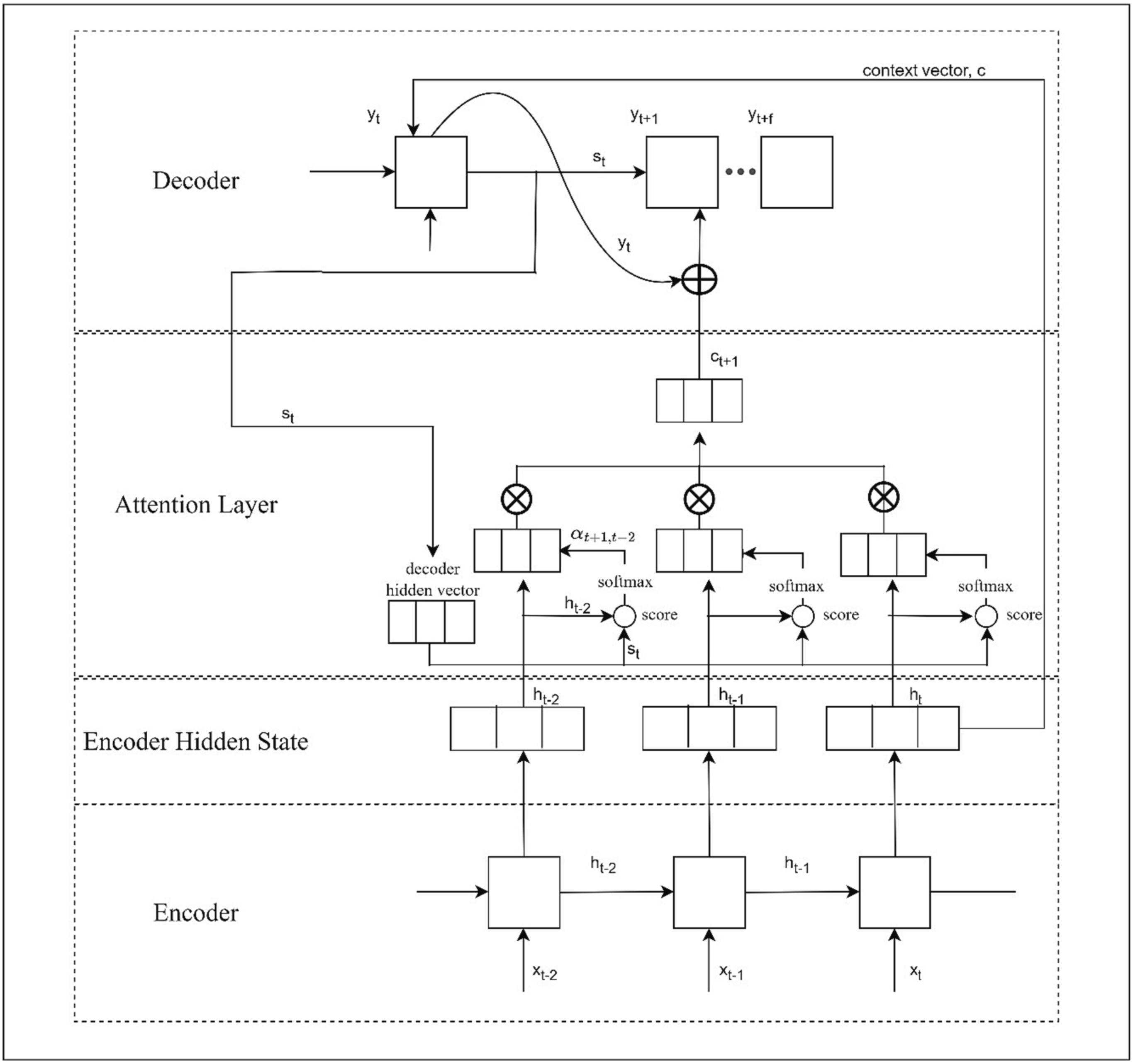

Fig. 5. A GRU encoder-decoder neural network with attention.

servations $\{y_{t+1}, y_{t+2}, \cdots, y_{t+n-1}\}$. In other words, the decoder defines joint conditional probability:

$$P(y_{t+n}|\{y_{t+1}, y_{t+2}, \cdots, y_{t+n-1}\}, c) = g(y_{t+n-1}, s_{t+n}, c_{t+n}) \quad (6)$$

where g is a non-linear function that outputs the probability of y_{t+n}, s_{t+n} is the hidden state of the decoder, and c_{t+n} is another context vector computed by the attention layer. The mechanism of the attention layer is explained below.

The context vector c_i from the attention layer is constructed using a sequence of hidden states $(h_1, h_2, ..., h_t)$, where $i \in [t+1, t+n]$. Each hidden state (i.e., h_t) contains the information on rainfall and runoff inputs at time step t. It is computed as a weighted sum of these hidden states:

$$c_i = \Sigma_{j=1}^{t} \alpha_{ij} h_j \quad (7)$$

and α_{ij} of each annotation h_j is computed by:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\Sigma_{k=1}^{t} exp(e_{ik})} \quad (8)$$

where $e_{ij} = a(s_{i-1}, h_j)$, and a is the alignment model that scores how well the input around position j and the output at position i match each other. If α_{ij} is a probability that the target runoff y_i at time step $i \in [t+1, t+n]$ is derived from source information of rainfall and runoff (i.e., x_j), then the i-th context vector c_i is the expected runoff over all the runoffs with probabilities α_{ij}.

Note that the probability α_{ij} or its associated e_{ij} scores reflects the importance of h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i (Bahdanau et al., 2014). Introducing this attention mechanism to the decoder eliminates the need to force the encoder to encode j-th runoff/rainfall information to the j-th hidden state. Instead, all the input information can be distributed across the hidden states by weights learned through attention.

3.2. Proposed GRU networks

In this study, we developed a general GRU-based Seq2seq Attention (GSA) model, as shown in Fig. 6, for the hourly streamflow (i.e., runoff) prediction of n hours in the future. The number n was determined by the average hydrograph at a particular sub-basin/location. The input features include (1) rainfall history and forecast of the current sub-basin, (2) streamflow history and future predictions of upstream sub-basins, and (3) streamflow history of the current sub-basin. The inputs were provided to the encoder layers, which were followed by a decoder layer with an attention mechanism. The architecture is then followed by a time series dense layer to predict the streamflow for n hours into the future. This DL model was applied hierarchically on the river network to predict streamflows for each sub-basin from upstream to downstream. Note that the sub-basins at the top layer of a river network don't have upstream sub-basins; thus, the corresponding features (i.e., upstream sub-basins' streamflow history and prediction) were set to zero.

For the neural network architecture (Fig. 6) employed in this study, two encoders were used to input time series with lengths. The first GRU encoder for the rainfall and upstream sub-basin streamflow observations has a length of $\tau_p + \tau_f$, and the second GRU encoder for the current sub-basin streamflow has a length of τ_p. The output contains a streamflow forecast for $n\Delta t$ future timesteps. For a downstream sub-basin, the streamflow forecasts of its upstream sub-basins will be helpful for the prediction. In this case, the upstream sub-basins refer to the immediate locations upstream of a particular sub-basin. Consequently, the DL model approximates a non-linear function, which represents the relationship between the input features and the output targets as;

$$q_{[t,t+n\Delta t]}^{<j>} = F\left(q_{[t-\tau_p,t]}^{<j>}, P_{[t-\tau_p,t+\tau_f]}^{<j>}, q_{[t-\tau_p,t+\tau_f]}^{<up>}\right) \quad (9)$$

where $<j>$ represents the current sub-basin, and $<up>$ is the sub-basin immediately upstream from j. The time duration between t_1 and t_2 is represented by $[t_1, t_2]$. Note that $\tau_p = 72$ hours and $\tau_f = 24$ hours were the maximum past and future time steps as mentioned in previous sections. In addition, q is streamflow, $\boldsymbol{F}$ is a non-linear function, and P is precipitation.

We investigated the effect of input features on model performance in this paper. Therefore, three variations are considered in addition to GSA, emphasizing the difference in input features, as shown in Fig. 7. The GSA_R model considers rainfall only. The input features were the past and future precipitation over the sub-basin. The GSA-RL model incorporates (1) the past and (2) future precipitation over the sub-basin as well as (3) the streamflow history at the current sub-basin's outlet. The GSA_RU model considers the histories and forecasts of rainfall and upstream sub-basins only. In other words, the input features of this model don't include the current sub-basin's streamflow history. Therefore, the input features include (1) the past and (2) future precipitation of the current sub-basin and (4) the past and (5) the future streamflow of upstream sub-basins. The GSA_R is unique in that the streamflow prediction for different sub-basins can be performed independently and simultaneously. Assuming there is no information on the rainfall forecast, we can simply set the future precipitation as zero in the original GSA model (GSA-ZFR). An alternative approach is a so-called GSA_RP model, in which the input features don't include the future precipitation of the current sub-basin.

The main goal of the project was to build a standard model pipeline that can be adapted to various basins and different locations. The model

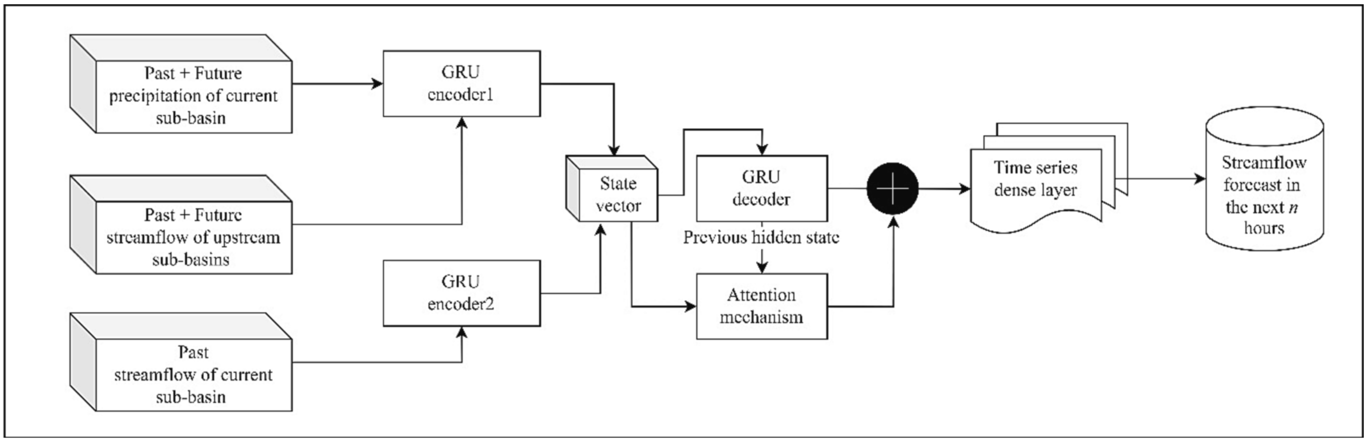

Fig. 6. A general GRU-based Seq2seq Attention (GSA) model for streamflow prediction.

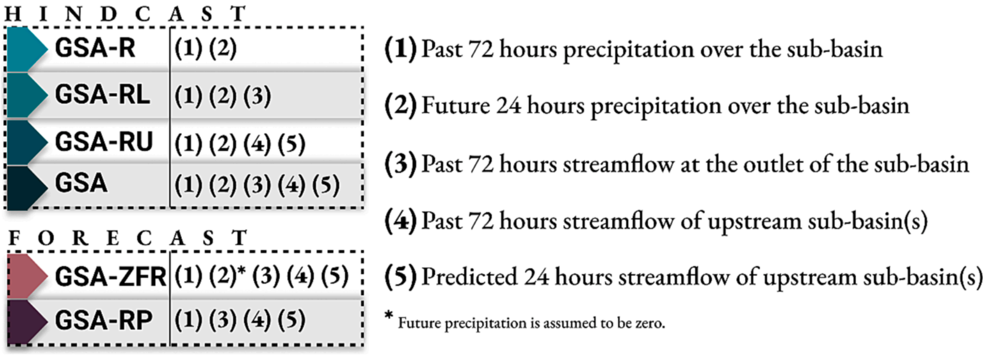
HINDCAST
GSA-R
(1) (2)
GSA-RL
(1) (2) (3)
GSA-RU
(1) (2) (4) (5)
GSA
(1) (2) (3) (4) (5)
FORECAST
GSA-ZFR
(1) (2)* (3) (4) (5)
GSA-RP
(1) (3) (4) (5)
(1) Past 72 hours precipitation over the sub-basin
(2) Future 24 hours precipitation over the sub-basin
(3) Past 72 hours streamflow at the outlet of the sub-basin
(4) Past 72 hours streamflow of upstream sub-basin(s)
(5) Predicted 24 hours streamflow of upstream sub-basin(s)
* Future precipitation is assumed to be zero.

Fig. 7. Variations of the GSA model with different input features.

settings and parameters we chose were mostly defaults of DL models in TensorFlow-Keras. The loss function in the proposed DL models is the mean squared error (MSE) between the true and predicted values, as defined by:

$$MSE = \frac{1}{N}\Sigma_{i=1}^{N}(y_i - Y_i)^2 \tag{10}$$

where N is the number of samples, y_i is the predicted value, and Y_i is the actual value.

3.3. Network training

The input variables (i.e., features) and the length of input sequences play an essential role in GRU encoders. As listed in Fig. 7, different DL models take different input features. The input features with past and future values have 96 time steps, consisting of $\tau_p = 72$ h for the past and $\tau_f = 24$ h for the future. However, the features with only past data have only 72 time steps. The length (n) of the output sequence is determined dynamically by taking half of the base length of an average hydrograph for a particular sub-basin in the river network. Consequently, both GSA and GSA_RP have two encoders (encoder1 and encoder 2) and one decoder, whereas GSA_R and GSA_RS have only one encoder (encoder 1) and one decoder. Each encoder or decoder layer has 512 GRU cells. After the decoder, only one dense layer outputs the predicted sequence. A linear activation function is used in the dense layer.

As discussed above, we conducted a total of 5,000 events, which were continuously simulated one by one for data collection. The last 1,000 events were used to generate the testing dataset for results and discussions in Section 5. The other 4,000 events provided the training and validation datasets with a 0.8/0.2 ratio to train the DL models with shuffling. The batch size chosen was 512 after experimenting with other sizes. The model is trained for 50 epochs every time an experiment is conducted. Adam optimizer was used as the optimization solver with an initial learning rate of 0.0001, which was reduced by a factor of 0.3 during the training process. The learning rate was reduced by a factor of 0.3. The models were developed based on Python 3 in the Keras framework with a TensorFlow backend, and 2 NVIDIA Quadro RTX 8000 GPUs were used in training.

4. Performance evaluation metrics and scale-independent benchmarks

4.1. Kling-Gupta Efficiency (KGE)

The metric we used for the performance evaluation of DL models is Kling-Gupta Efficiency (KGE), a commonly used statistic for evaluating hydrologic models. KGE is computed as,

$$KGE = 1 - \sqrt{(r-1)^2 + \left(\frac{\sigma_f}{\sigma_o} - 1\right)^2 + \left(\frac{\mu_f}{\mu_o} - 1\right)^2} \tag{11}$$

where r is Pearson's correlation, σ_f is the standard deviation in predictions, σ_o is the standard deviation in observations, μ_f is the mean of predictions, and μ_o is the mean of observations (Gupta et al., 2009). The value of KGE ranges between 1 and $-\infty$. The ideal KGE value is 1, representing a perfect fit between predictions and observations. There is no specific KGE threshold in the literature for "good" or "bad" model performance. Positive values of KGE, however, are generally accepted as a sign of "good" performance (Knoben et al., 2019). As broadly discussed by Knoben et al. (2019), $KGE \cong -0.41$ is equivalent to using the mean of observations as a predictor. The use of KGE is subject to possible biases highlighted by Clark et al. (2021) and Lamontagne et al. (2020).

To assess the model performance, we calculated the KGE individually for each rainfall-runoff event (i.e., 20-day duration) in the test dataset and then calculated the median KGE score to indicate overall model performance. In calculating a KGE statistic for a single event, we only consider the direct runoff component of the hydrographs, from the beginning of the rising limb to the end of the recession process, excluding baseflow from the evaluation procedure. This is because we specifically aim to quantify the flood prediction ability of the proposed DL models and the contribution of input information in predicting the direct runoff resulting from a storm event. Thus, considering our intentions, including baseflow in model evaluation might be misleading because it always exists in the river system and dominates streamflow time series most of the time (Krajewski et al., 2021).

4.2. Temporal persistence

We used temporal persistence for benchmarking the prediction skill of the DL models introduced in this study due to its simplicity and functional utility. The temporal persistence approach has been comprehensively discussed in (Ghimire and Krajewski, 2020; Krajewski et al., 2021, 2020) and used by some recent studies as a reference to assess AI-based streamflow predictions (Sharma et al., 2023; Xiang et al., 2021). Krajewski et al. (2021) recommended that any data-based models should be judged based on persistence rather than simply showing performance metric values achieved by data-based models. Using streamflow persistence as a benchmark method allows us to evaluate the degree of improvement offered by DL models.

Temporal persistence relies on the concept of "tomorrow will be like today". One, for example, can assume that the streamflow at time $t+\Delta t$ will be the same as the one at the time of observation t, i.e., $q(t+\Delta t) = q(t)$. The advantage of temporal persistence is that the flood peaks are predicted accurately but with a Δt lead time. However, the skill of such predictions diminishes as the lead time increases, and this is highly correlated with the scale of the basin of interest (Ghimire and Krajewski, 2020; Krajewski et al., 2020). For basins with greater drainage area, the skill of persistence forecast at a specific lead time would outperform the forecasts for smaller basins. In the Turkey River case, for instance, persistence forecasts at a lead time of 24 h at Spillville would show poor performance compared with the forecasts at Garber.

This phenomenon raises the question of how to evaluate model performance irrespective of the scale of the basins. To address this issue, we introduced the concept of scale-independent time (i.e., a dimensionless time), which can be interpreted to mean that predicting at a particular future reference time is equally difficult for any basin scale. The equation for dimensionless time is as follows:

$$T^{*} = \frac{T_{lead}}{T_{(KGE=0)}} \tag{12}$$

where T_{lead} is a particular lead time for prediction, and $T_{(KGE=0)}$ is the reference time when the KGE becomes zero at the basin of interest. Our experimental studies with sinusoidal wave-shaped synthetic time series and exploratory analysis with the data used in this study indicated that the value of *KGE* is dominated by the correlation coefficient term in Eq. (11). In fact, $KGE = 0$ corresponds to zero correlation between the time series compared (see Supplementary Fig. 1). However, while this assumption holds true within the context of this study, it may not universally apply to real-world applications.

In Fig. 8, we present the median KGE scores for persistence at the five "gauged" basins in the Turkey River Basin for different lead times up to four days. As clearly indicated in the top plot of Fig. 8, the skill of the persistence model is notably associated with the basin scale: The decay in the performance of persistence-based predictions with increasing lead times is relatively slow in Garber, compared to the inner locations. The persistence model for Spillville performed worst among all "gauged" locations in our virtual basin since it has the smallest drainage area. When all the locations are considered, the KGE value for the 20-hour lead time ranges between 0.2 and 0.65. For each sub-basin, we assume the lead time at which the KGE value for persistence corresponds to zero as the reference time. We provide site-specific reference times in Table 1, which can also be inferred from the top plot in Fig. 8. Note that when we plot the KGE scores for persistence on a dimensionless time axis in the bottom plot of Fig. 8, the lines depicting persistence prediction skills for the "gauged" basins get closer to each other and almost overlap. Using KGE scores greater or smaller than zero to determine the reference time will result in distinct separations of the persistence prediction lines.

Throughout the study, we use the average persistence performance depicted by the black dashed line in Fig. 8 as the benchmark. Moreover, we employ the notions of near, intermediate, and far future, each of which describes a range of dimensionless time, to better convey the results and assess the proposed DL models. The near future is constrained to values smaller than 0.1 on the x-axis, while the intermediate future is bounded to the range between 0.1 and 0.5. The far future takes any dimensionless time values greater than 0.5, extending until the end of the investigated range of lead time (which roughly corresponds to 2.5 dimensionless time units).

Table 1
Area and reference time for each sub-basin.

	Spillville	Eldorado	Elkader	Littleport	Garber
Area [km^2]	458	1667	2359	909	4031
Reference Time [hour]	25.0	31.0	43.0	32.0	44.0

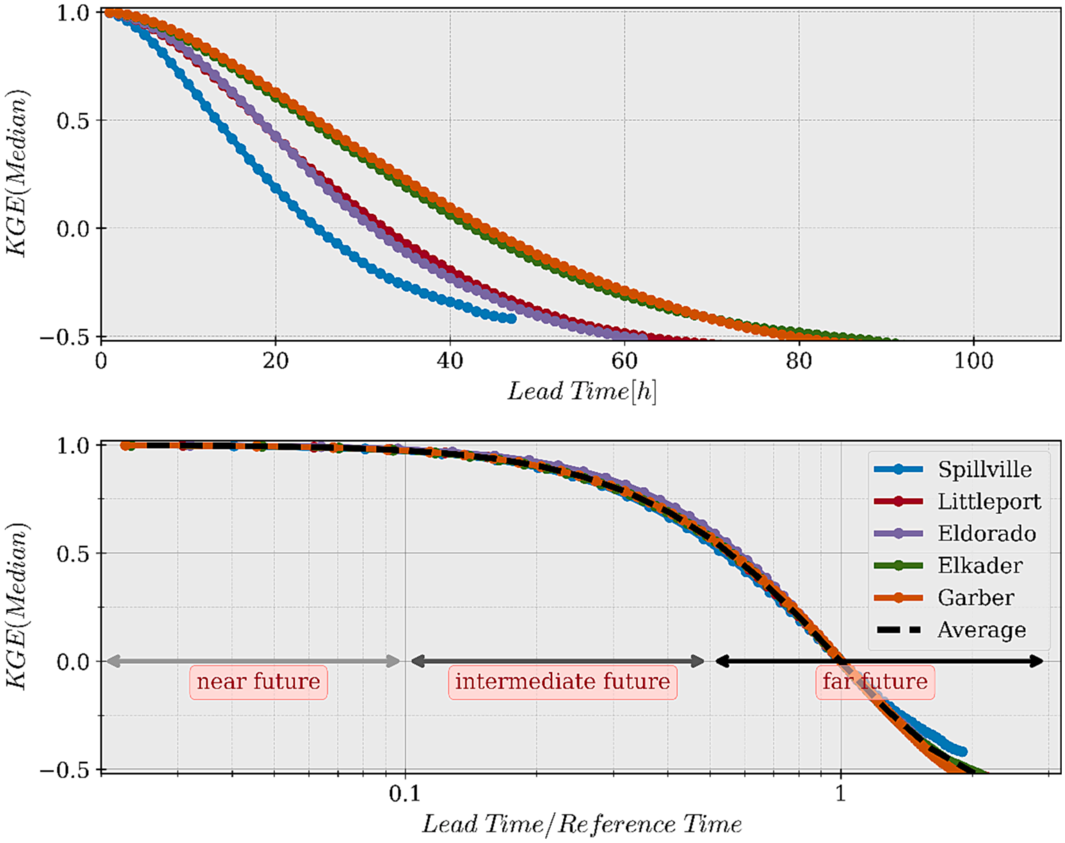

Fig. 8. Performance of the temporal persistence method in "gauged" basins in Turkey River. Top: Performance of persistence-based predictions across various lead times. Bottom: The same information plotted against scale-independent values on the x-axis (dimensionless time). These values are computed by assuming the lead time corresponding to KGE = 0 as the reference time for each sub-basin.

5. Results and discussion

This section is divided into three subsections. First, we present results on the streamflow predictability skills of our four proposed DL models: GSA-R, GSA_RL, GSA_RU, and GSA as described in Fig. 7, only within the investigated range of lead time. These four models implement the Hindcast Mode strategy, meaning precipitation in the interval $[t, t + \tau_f]$ is a known input from rainfall predictions. Given different input time-series data sets, we can interpret these models as a response function for the sub-basins. The purpose of this exercise is to evaluate various input sources. Second, we use the GSA with zero future precipitation (i.e., GSA-ZFR) and GSA-RP to test the capabilities of DL models in Forecast Mode, which means that rainfall in the interval $[t, t + \tau_f]$ is unknown. Third, we study the influence of training dataset size on prediction performance using the GSA model. We also explore the uncertainty in the model's prediction by experimenting with various instances of small datasets.

In the first and second subsections, we independently analyze the DL models and report model performance for all five "gauged" basins in the virtual environment. We aim to comprehend how the predictive abilities of the models change across scales. The last section, however, considers the average model performance involving all analyzed locations since the purpose here is to learn about the effect of dataset size on overall model performance.

Throughout the results section, we use the persistence of streamflow as the baseline and assess the predictive performances of various DL models. If, for instance, the KGE score of a model for a given dimensionless time is greater than the KGE score of temporal persistence, we consider the model performance acceptable. We emphasize that the persistence line in the performance plots represents the average persistence model skill of all analyzed sites, thanks to the normalization strategy described in section 4.2.

5.1. *Performance of DL models in Hindcast Mode*

Fig. 9 illustrates the KGE statistic for the GSA-R, which is the simplest model in terms of the information used as input and uses only the rainfall history and the perfect future rainfall information for the basin of interest. The model produces a median KGE greater than 0.5 for all sub-basins, with a minor rise as lead time increases. The model performs best in Littleport and Spillville, reaching KGE values over 0.75 for nearly all lead times considered, while its performance in Garber and Elkader is slightly lower. The figure shows that the GSA-R's performance improves when the drainage-area of the basin at which predictions are made decreases. In addition, the GSA-R is a better predictor than the benchmark metric, simple temporal persistence, in the far future but not in the near and intermediate future. In other words, the model is better at identifying a response function to transform rainfall into runoff at longer lead times. The temporal persistence method, however, is a superior choice to predict streamflow for shorter lead times.

To understand the role of local streamflow information in streamflow prediction, we add local streamflow data to the input features used by GSA-R and train another model called GSA-RL. We present the median KGE scores for this model in Fig. 10. A KGE of greater than 0.75 is accomplished at all "gauged" sites regardless of the lead time. In fact, the performance of the model remains at a steady level irrespective of the location and lead time. Among all basins, Garber is the one where the GSA-RL reaches the highest KGE value, which is about 0.85. Spillville, which has the smallest drainage area, consistently maintains a KGE of around 0.75 at all lead times. Even though there are no quantitatively profound differences in the model's performance with respect to the sub-basins, it is noticeable that the model performs best at sub-basins with a greater upstream area. The results suggest that the predictive performance of the model reduces as the drainage area decreases. Compared with the GSA-R model, GSA-RL showed better performance, especially at downstream sites. Although the improvement in the model performance is limited at exterior sub-basins, the inclusion of local streamflow information into the input features improves the model's streamflow prediction ability. This is particularly true at locations with a greater upstream area, as shown in Fig. 10. GSA-RL, like GSA-R, outperforms the temporal persistence method in the far future, although the benchmark metric continues to excel in the near and most of the intermediate future.

We then develop the GSA-RU model on top of the GSA-R. This model uses the upstream streamflow data as an additional input feature. Fig. 11 depicts the median KGE statistics for the GSA-RU at all the "gauged" sites individually, along with the average performance of temporal persistence. The figure shows that the model performs decently with a KGE value near 0.75 at all locations and lead times. The highest KGE values are obtained at Garber; however, the model's performance was poorer in exterior sub-basins. Compared to the GSA-R, the GSA-RU significantly improves prediction accuracy at Garber and Elkader while showing little or no gain at Spillville, Littleport, and Elkader. The performances of GSA-RU and GSA-RL, however, are comparable. Fig. 11 also indicates that the GSA-RL outperforms the temporal persistence in the far future, while the temporal persistence method is superior in the near and intermediate future.

The final model employing the Hindcast Mode Strategy is the GSA,

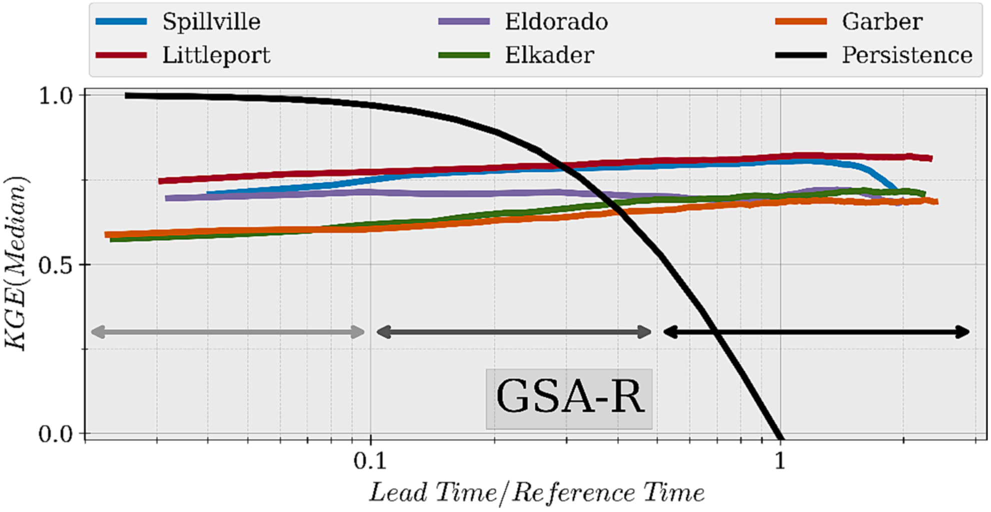

Fig. 9. The median KGE for the GSA-R model at the gauged basins.

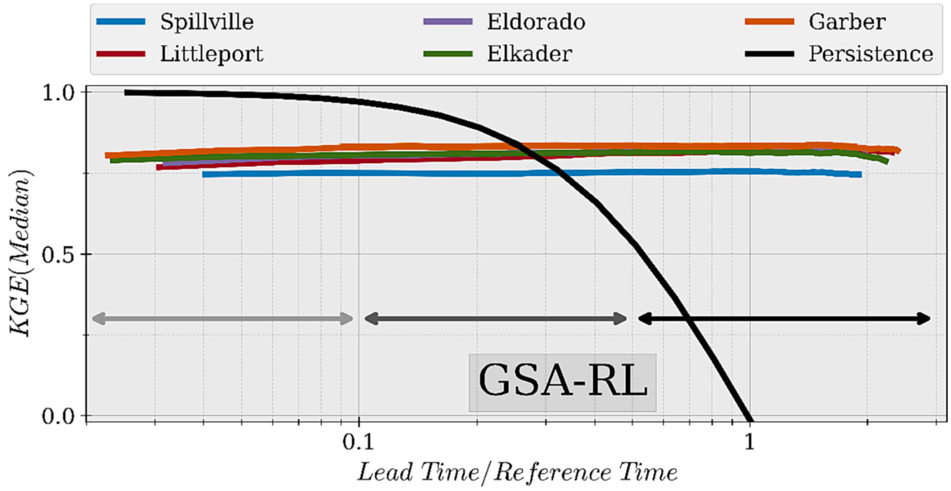

Fig. 10. The median KGE for the GSA-RL model at the gauged basins.

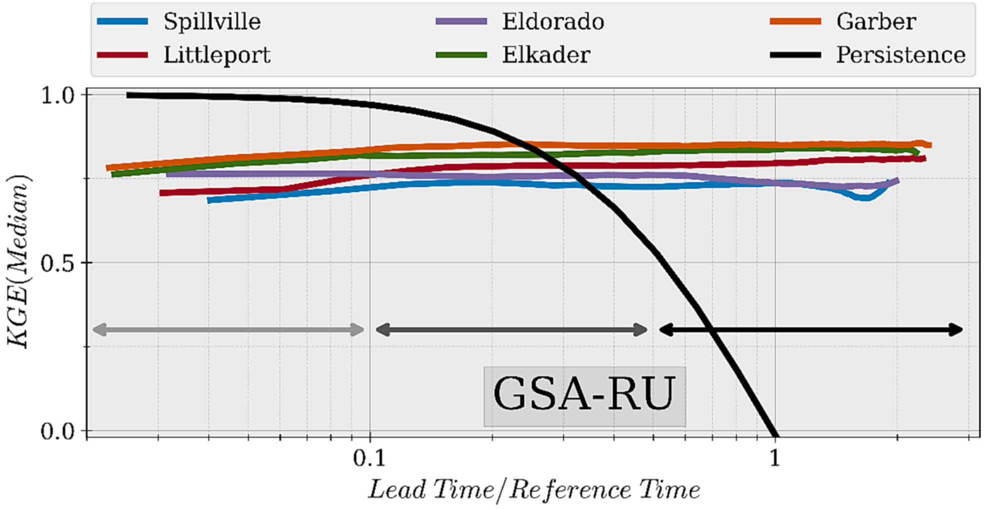

Fig. 11. The median KGE for the GSA-RU model at the gauged basins.

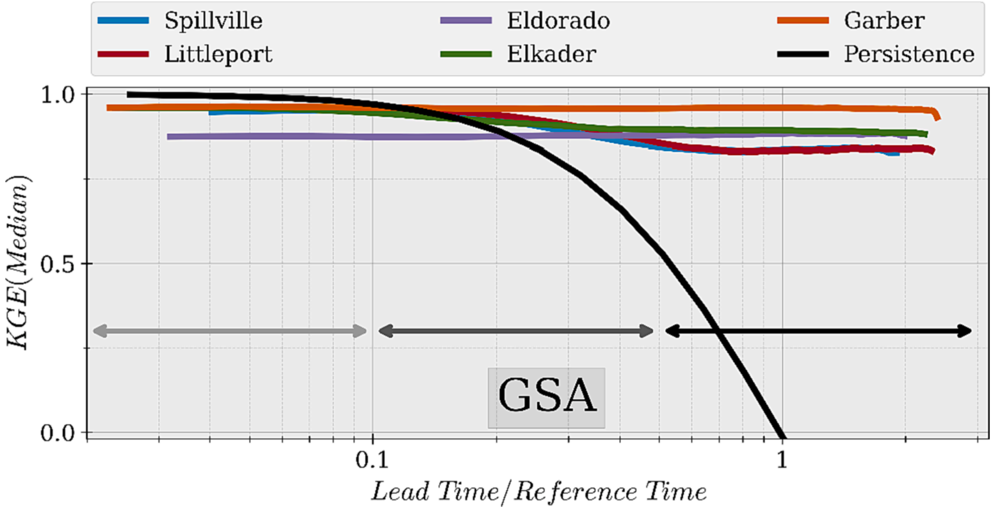

Fig. 12. The median KGE for the GSA model at the gauged basins.

which exploits all available information, including past/future rainfall and local streamflow data as well as the upstream streamflow data (see Fig. 7). We present the KGE scores for this model in Fig. 12. As anticipated, the GSA shows higher accuracy at every sub-basin and lead time compared to the other models evaluated in this section. A median KGE close to 1 is achieved at Garber for all the lead times considered. With the exception of Eldorado, the model performs similarly for the remaining sub-basins, with a median KGE greater than 0.9 in the near future and a downward trend in KGE statistics throughout the intermediate future, followed by KGE values well over 0.75 in the far future. When measured against the temporal persistence method, the GSA model is a better predictor in both the intermediate and far future. This makes it the best of the four Hindcast models that we tested.

The results consistently showed that the temporal persistence outperforms the DL models in the near future. This observation gains particular interest when considering DL models like GSA_RL and GSA, which utilize local streamflow information as an input feature. The reason for the underperformance of these DL models in the near future is that each DL model is individually generalized for the Turkey River Watershed, encompassing both temporal and spatial dimensions. In other words, each model is trained to minimize losses within the prediction horizon across all considered sub-basins. Therefore, while the DL models achieve a balance between prediction performance in near and far futures, persistence excels specifically in the near future.

Our results confirm the conclusions and results put forward by multiple studies that have used DL models (LSTMs, GRU, etc.) to identify the relation between rainfall over a basin and streamflow fluctuations at the outlet. We obtained similar KGE values to those reported in the literature (Demir et al., 2022; Xiang et al., 2021). In Fig. 12, we show that if perfect information of past and future rainfall and past local streamflow are provided to the DL model (i.e., the GSA model), the performance of predicting future streamflow fluctuations or the response of the catchment to a rainfall event is very high (KGE values are between 0.80 and 0.95). It shall be noted that perfect fitting cannot be achieved due to inherent data uncertainty. Such uncertainty was introduced because spatially variable rainfall is employed in the process-based model (i.e., HLM), a factor not accounted for in the training data which only includes basin-average rainfall information. As a remarkable result, we observe that the basin-average rainfall information is essential for accurately producing streamflow hydrographs in our study and achieving accurate predictions. Furthermore, the streamflow hydrographs calculated for our virtual basin were produced by solving a large set of differential equations to calculate the time variability of mass transfer from hillslopes and channel links toward the basin outlet. Each control volume was externally forced with unique rainfall and evaporation time series. Conversely, the DL models used time series of hourly averaged rainfall as input and streamflow values at the outlet to predict future streamflow fluctuations.

5.2. Performance of DL models in Forecast Mode

As shown in Fig. 13, GSA-ZFR (i.e., the model with zero future precipitation) achieves median KGE values close to 1 in the near future at all sites in the virtual environment. The overall impact of inaccurate future rainfall estimates (i.e., zero future rainfall), aside from instances where a zero-rainfall forecast is accurate, becomes apparent by the end of the intermediate future with a decreasing performance at all locations considered. This is followed by a more pronounced drop in model performance in the far future. Although forcing the trained GSA model with zero future rainfall (i.e., GSA-ZFR) does not affect the accuracy of the predicted hydrographs in the near future, it results in a significant reduction in the model performance with increasing lead times. In addition, the reduction in model performance is highly location dependent. While the model achieves a KGE greater than 0.75 in Garber, it is well below zero in Spillville at the highest value of the ratio of lead time to reference time. It can be inferred from Fig. 13 that larger sub-basins show less of a decline in model performance as lead time increases. A comparison of the GSA-ZFR and the benchmark metric (i.e., temporal persistence) reveals that, even with the false rainfall estimates, the GSA model outperforms the temporal persistence method in the intermediate and far future but not in the near future.

The performance of GSA-RP, which excludes future rainfall information from the input features, is shown in Fig. 14. GSA-RP achieves a median KGE over 0.75 at all locations in the near and intermediate future. Although the model maintains a constant KGE score until the end of the intermediate future, there is a considerable drop in model performance in the far future. The KGE values drop below zero in Spillville, Littleport, and Eldorado by the end of the prediction horizon, while they stay between 0 and 0.25 in Garber and Elkader. Although the better performance of GSA-RP in locations with larger drainage areas is noteworthy, the dependence of model performance on basin scale was not as pronounced as was observed for GSA-ZFR.

During the latter half of the intermediate future, GSA-RP begins to demonstrate a higher KGE than the benchmark persistence method. However, GSA-ZFR and GSA-RP both fail to provide accurate predictions at longer lead times, especially when compared with GSA, which takes advantage of all available information, including past and perfectly known future rainfall. Nevertheless, the models in forecast mode are still

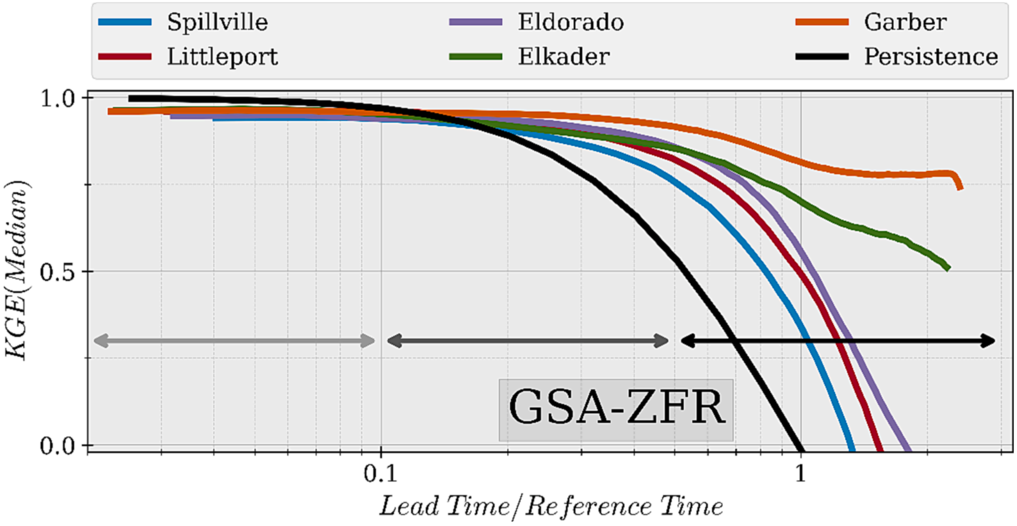

Fig. 13. The median KGE for the GSA-ZFR with zero future rainfall at the gauged basins.

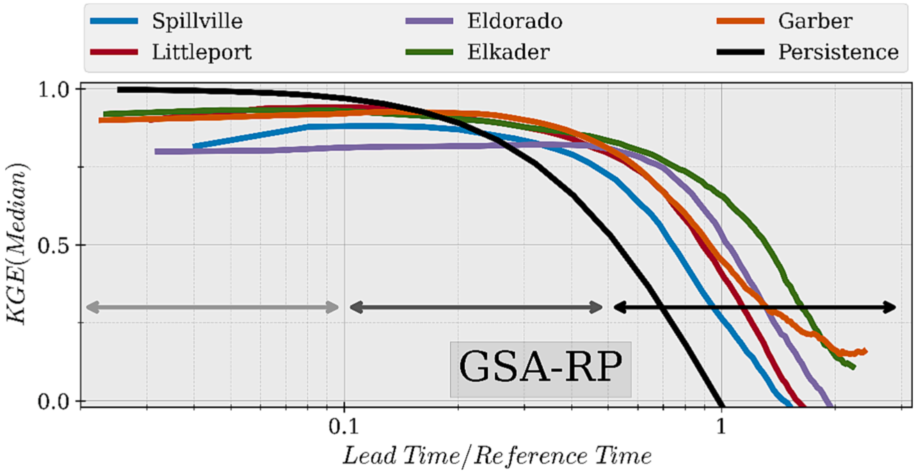

Fig. 14. The median KGE for the GSA-RP model at the gauged basins.

valuable in the near and intermediate future.

Our experimental setup has allowed us to evaluate the performance of DL models in both Hindcast and Forecast modes. For hindcast evaluation, the models were trained using a perfect knowledge of past and future rainfall, while in the forecast evaluation, the models were used and later retrained without knowledge of the actual values of future rainfall. These are the two extreme scenarios that can be found in real-time forecasting. Furthermore, we show that the Forecast DL models, which do not receive accurate knowledge of future rainfall, demonstrate a rapid decay in their ability to predict future streamflow for all the considered scales. The Forecast DL models are only slightly better than the persistence benchmark.

5.3. *Effect of dataset size on deep learning training*

In this study, we used synthetically generated rainfall events to produce streamflow data in our virtual environment. This gives us the power to generate as much data as we wish. In real-world applications, however, the length of streamflow data is limited to the observation period at desired locations. Therefore, instead of using a dataset including thousands of events, we explored the use of smaller data samples to assess the uncertainties in the predictions provided by the GSA model in our virtual environment. The above exercises use a dataset comprising 5000 events to train and test the DL models. As mentioned, we followed an 80:20 split strategy to create training and test datasets, meaning that 4000 events were used to train the models, and 1000 were used to test them. It is reasonable to anticipate that the dataset size will affect the ability of AI models to transform rainfall into runoff. This assertion needs to be investigated by performing analyses with datasets of varying sizes.

The goals of the exercise in this section were to explore the influence of the training dataset size on streamflow prediction and to understand the extent of such impact or dependency. To this end, we use the GSA architecture, exploiting all available information at the location of interest, knowing that it provides the most accurate streamflow predictions. We also use the average model performance in exploring the influence of dataset size rather than focusing on how the models trained with various amounts of data behave across scales. We consider 100, 500, and 2000 events as well as 5000 events whose results were presented in section 5.1. It shall be noted that we do not randomly select individual events when sampling data from the complete dataset of 5000 events. Instead, we randomly select consecutive events to have smooth time series data. In each case, 80 % of the data is used for training with an 80/20 split for validation, and 20 % is used for testing (i.e., calculating the KGE statistics) to assess the GSA model. For example, in the case of 100 events, the training (including validation) and testing datasets comprise 80 and 20 events, respectively.

Fig. 15 illustrates the performance of the GSA model for the cases of 100, 500, 2000, and 5000 events. The median KGE scores are above 0.75 regardless of the dataset size used for model training. The model's performance when using a dataset of 2000 events is almost identical to that observed when a dataset of 5000 events is used. A decrease in

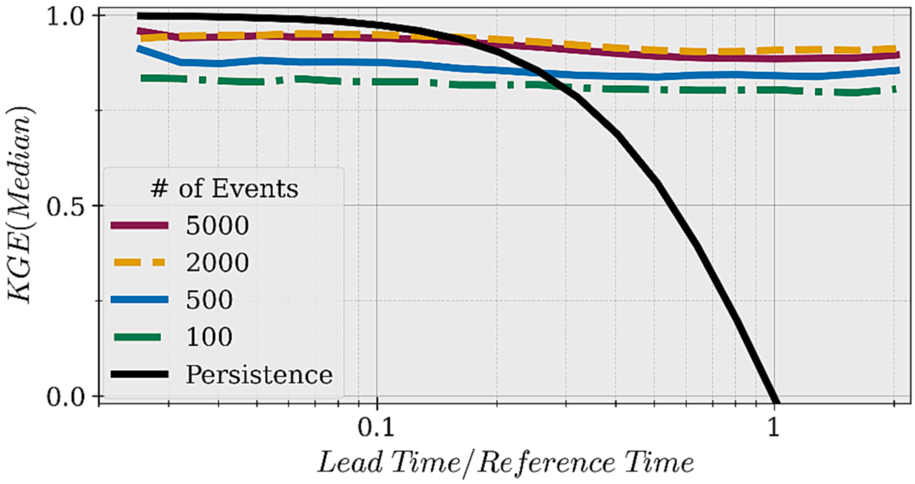

Fig. 15. The performance of the GSA model when various rainfall-runoff events are used for training and testing.

prediction performance becomes noticeable for a sample size of 500 events. However, the model still performs well for all lead times considered in this study, even when the number of events is reduced to 100. Overall, these results suggest that the larger the training dataset's size, the better the model performs. However, there appears to be a limit to the improvement that can be achieved for model performance by increasing the size of the training dataset. In this study, 2000 events were sufficient to build a model that performs as well as one trained using 5000 events.

Considering the typical observation period of readily available streamflow data worldwide, we create 50 distinct sets or realizations of 100 events, which generally equates to 30 years of streamflow record with the assumption that 3 or 4 flood events occur each year. The GSA model was trained using 80 events (with a validation split of 20 %) for each of these realizations. Then, the model performance was evaluated using the remaining 20 events.

The results of these experiments are given in Fig. 16. In the figure, each grey line indicates the GSA model trained and evaluated for an individual realization. The upper (red line) and lower bounds (blue line) were obtained by selecting the maximum or minimum median KGE among all realizations at specific lead times. The mean line depicted by the black dashed line corresponds to the mean of all realizations. The results point out that the model's performance varies with respect to the data sample used for training and testing. The figure shows that the upper bound exceeds a median KGE value of 0.8 and does not indicate any decrease with increasing lead times. The lower bound, however, starts with a KGE value slightly below 0.75 and decreases up to about 0.5 at the longest lead time considered in this study. It is evident in the figure that the uncertainty boundary becomes broader as the lead time or the ratio of lead time to reference time increases.

We conducted a series of experiments to determine how much data (measured as a total number of rainfall events) is needed to achieve the performance exhibited by the GSA model when it is trained using 4,000 events. We train the same DL model with samples of 1600, 400, and 80 events, and we demonstrate (as expected) that the performance is reduced but still maintains significant levels of accuracy (KGE > 0.75) even in cases where only 80 events were used. We further show that the performance achieved with fewer events (e.g., 80 events) depends on the specific samples selected for training and validation. In Fig. 16, we presented results indicating that after training DL models using small data samples of only 80 events, the performance using KGE can be as low as 0.75. This provides a real benchmark for real-world applications where the number of sampled events can be relatively small.

6. Conclusions

A process-based model typically involves solving numerous differential equations, which can range from hundreds to thousands, depending on the scale of the basin. Consequently, computational intensity emerges as an imperative concern that demands attention in the realm of process-based modeling, particularly when subject to precipitation uncertainties. Conversely, when provided with real data, a process-based model may manipulate its model parameters to replicate diverse streamflow scenarios. However, when applied to dissimilar basins or varying rainfall patterns, these derived parameters might lack applicability.

In contrast, deep learning, as a data-driven approach, can be trained to forecast streamflow or other flood-related variables once an appropriate dataset is available. Although the training phases for DL models can be time-consuming, the subsequent predictions are faster than process-based models. This advantage becomes particularly pronounced when dealing with extensive real datasets. Moreover, a pre-trained DL model, trained on data from various basins, can be fine-tuned to suit a specific basin or rainfall event via transfer learning. Nevertheless, it is essential to note that the quality of the data significantly influences the performance of DL models when tested on new, unseen data. Prior research has demonstrated remarkable achievements by DL models but trained with high-dimensional and extensive spatiotemporal real data (Kratzert et al., 2019a, 2019b; Mai et al., 2022).

Deep learning algorithms are gaining recognition in water resources engineering as a tool capable of predicting different aspects of the hydrologic cycle. In this paper, we test the ability of DL models to predict rapid streamflow fluctuations following significant rainfall events. We use a virtual basin to test the DL capabilities, which allows us to remove performance issues associated with the uncertainty of rainfall and streamflow estimates when working with actual observations. In addition, the virtual basin allows us to control the complexity imposed in the rainfall-runoff and transport equations, and it serves as a benchmark for the kind of performance that can be expected when high-quality data are available to train DL models. Our results indicate that DL models are powerful tools to determine the input–output function connecting streamflow fluctuations to rainfall inputs, especially in the intermediate and far futures.

The benchmark used in this study to evaluate DL models is local persistence, meaning that the streamflow in a future moment is assumed to be the same as the current streamflow. This is an ideal metric to determine the value of a model because it uses intrinsic information for the time series under consideration. The KGE for the persistence-based prediction decays at different rates for basins of various sizes. However, this study shows that the systematic decay can be scaled into a single

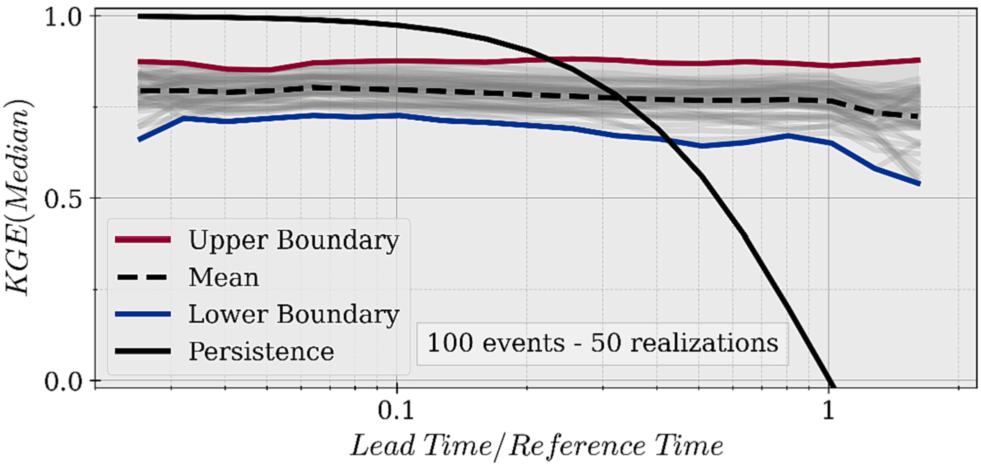

Fig. 16. The uncertainty bounds of KGE values for 50 realizations of the GSA model. Each realization contains 100 rainfall-runoff events.

function for all basins using the lead time at which KGE equals zero for each specific sub-basin. This collapse of different decay curves of KGE, shown in Fig. 8, allows us to make generic statements about the performance of the models tested that are valid for sites draining different basin sizes and with varying degrees of upstream information. The results presented in the plots use a dimensionless time axis (Lead Time/ Reference Time) and can be turned into specific predictions for a site of interest using the values in Table 1.

We show that the DL models provide a consistent level of performance across all sites and for all relative lead times when past and future rainfall is provided as input. However, the performance for longer lead times decreases quickly when future rainfall is removed as a known input. This difference in performance indicates that DL is identifying a non-linear response function for the basin, analogous to the unit response function (or unit hydrograph) that hydrologists have identified and used for many decades to connect effective rainfall to streamflow variations. The response function identified by DL goes several steps further than classical hydrograph analysis because it also simultaneously identifies the correct rainfall-runoff transformation and the non-linear basin response function. The former is what portion of the precipitation becomes runoff, while the latter results from travel along the channels in the river network, i.e., resulting from routing equations and parameterizations of hydraulic geometry that were assumed. Therefore, the value of DL models in a Forecast Mode environment will depend very strongly on how well future precipitation can be predicted. In view of these results and previous modeling experiences (Ghimire et al., 2021; Quintero et al., 2020; Velásquez et al., 2021), we hypothesize that DL models are more sensitive to the precision of future rainfall estimates than physics-based distributed models because the latter benefit from integrating water flows in space and time rather than through an imposed response function. Testing this hypothesis will be the subject of a future study.

We also used our diagnostic setup to test the size effect of the training set on the performance of the DL model. Such analyses were aimed at benchmarking expected performance on a realistic dataset where the number of events available in a dataset is more likely to be in the hundreds of events than in the thousands (a typical streamflow time series for basins of the size considered in this study may contain 3 to 4 events per year and records are typically 30 years long). We show that the performance of the algorithms changes significantly when only hundreds of events are used in comparison to thousands of events. In addition, the DL models demonstrated variations in performance that depended on the specific dataset of a given size that was chosen for a particular test. We conclude that the variability in performance depends on how much hydrograph variability is captured by the training set relative to the range of existing hydrographs in the validation set.

As applications of ML and DL become more commonplace in hydrology, we recommend using diagnostic controlled experiments that can provide appropriate benchmarks and performance tests for these new methods. Also, a diagnostic test can provide guidance and limitations for applying DL models in cases with limited data or where physical conditions may prevent their applicability. Our work here suggests that future research will need to be done to investigate how complexity in the processes that occur in a catchment or the temporal and seasonal variability of dominant processes and limit the applicability of ML to identify consistent response functions. We also recommend using event-specific and locally relevant performance metrics that measure specific aspects of streamflow fluctuations. We can quantify differences in performance metrics that do not include baseflow conditions in the time series and propose a dimensionless axis for lead time to inform the performance of the streamflow time series in a way that is independent of the upstream basin area. Controlled experiments using distributed hydrological models provide a valuable scenario to test new and existing data analysis tools before they are applied to real data where all the complexities of the hydrological system, uncertainty in observations, and heterogeneities in the landscape, occur simultaneously.

To sum up, in this study, we employed simulation data to establish a benchmark for evaluating DL models' performance and investigate how various input features and training data sizes impact their performance. As we move forward, our research will pivot towards the utilization of authentic datasets, focusing intently on data collection strategies and meticulous feature selection..

CRediT authorship contribution statement

Faruk Gurbuz: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Avinash Mudireddy:** Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Ricardo Mantilla:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision. **Shaoping Xiao:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgements

We want to thank the support and funding from the Iowa Initiative for Artificial Intelligence (IIAI) and the Iowa Flood Center. We would like to extend our thanks to the reviewers for their invaluable feedback and constructive suggestions. Xiao would like to acknowledge the support from the US Department of Education (ED#P116S210005) and the US National Science Foundation (grant #2226936). Mantilla also acknowledges the support from the Natural Sciences and Engineering Research Council of Canada RGPIN-2023-04724). The codes used are available in a GIT repository under the MIT open-source license terms at https://github.com/Avi1122/GRU-based-Seq2seq-Attention-Model.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jhydrol.2023.130504.

References

Ayzel, G., Heistermann, M., 2021. The effect of calibration data length on the performance of a conceptual hydrological model versus LSTM and GRU: a case study for six basins from the CAMELS dataset. Comput. Geosci. 149, 104708 https://doi.org/10.1016/J.CAGEO.2021.104708.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural Machine Translation by Jointly Learning to Align and Translate. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179.

Clark, M.P., Vogel, R.M., Lamontagne, J.R., Mizukami, N., Knoben, W.J.M., Tang, G., Gharari, S., Freer, J.E., Whitfield, P.H., Shook, K.R., Papalexiou, S.M., 2021. The abuse of popular performance metrics in hydrologic modeling. Water Resour. Res. 57 https://doi.org/10.1029/2020WR029001.

Demir, I., Xiang, Z., Demiray, B., Sit, M., 2022. WaterBench-Iowa: a large-scale benchmark dataset for data-driven streamflow forecasting. Earth Syst. Sci. Data 14, 5605–5616. https://doi.org/10.5194/essd-14-5605-2022.

England, J.F., Julien, P.Y., Velleux, M.L., 2014. Physically-based extreme flood frequency with stochastic storm transposition and paleoflood data on large watersheds. J Hydrol (amst) 510, 228–245. https://doi.org/10.1016/j.jhydrol.2013.12.021.

Estévez, J., Bellido-Jiménez, J.A., Liu, X., García-Marín, A.P., 2020. Monthly precipitation forecasts using wavelet neural networks models in a semiarid environment. Water (switzerland) 12, 1909. https://doi.org/10.3390/W12071909.

Fonley, M.R., Qiu, K., Velásquez, N., Haut, N.K., Mantilla, R., 2021. Development and evaluation of an ODE representation of 3D subsurface tile drainage flow using the HLM flood forecasting system. Water Resour. Res. 57, e2020WR028177 https://doi.org/10.1029/2020WR028177.

Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M., Lin, Q., 2020. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. J Hydrol (amst) 589, 125188. https://doi.org/10.1016/J.JHYDROL.2020.125188.

Ghimire, G.R., Krajewski, W.F., 2020. Exploring persistence in streamflow forecasting. JAWRA Journal of the American Water Resources Association 56, 542–550. https://doi.org/10.1111/1752-1688.12821.

Ghimire, G.R., Krajewski, W.F., Quintero, F., 2021. Scale-dependent value of QPF for real-time streamflow forecasting. J. Hydrometeorol. 22, 1931–1947. https://doi.org/10.1175/JHM-D-20-0297.1.

Granata, F., Gargano, R., de Marinis, G., 2016. Support vector regression for rainfall-runoff modeling in urban drainage: a comparison with the EPA's storm water management model. Water (basel) 8, 69. https://doi.org/10.3390/w8030069.

Gruber, N., Jockisch, A., 2020. Are GRU cells more specific and LSTM cells more sensitive in motive classification of text? Front Artif Intell 3, 40. https://doi.org/10.3389/frai.2020.00040.

Gupta, H.V., Kling, H., Yilmaz, K.K., Martinez, G.F., 2009. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. J Hydrol (amst) 377, 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003.

Ha, S., Liu, D., Mu, L., 2021. Prediction of Yangtze River streamflow based on deep learning neural network with El Niño–Southern Oscillation. Scientific Reports 2021 11:1 11, 1–23. https://doi.org/10.1038/s41598-021-90964-3.

Haykin, S.S., 1999. Neural networks : a comprehensive foundation. Prentice Hall.

Heiss, W.H., McGrew, D.L., Sirmans, D., Heiss, W.H., McGrew, D.L., Sirmans, D., 1990. Nexrad - Next generation weather radar (WSR-88D). MiJo 33, 79.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

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 (basel) 10, 1543. https://doi.org/10.3390/w10111543.

Knoben, W.J.M., Freer, J.E., Woods, R.A., 2019. Technical note: Inherent benchmark or not? comparing nash-sutcliffe and kling-gupta efficiency scores. Hydrol. Earth Syst. Sci. 23, 4323–4331. https://doi.org/10.5194/hess-23-4323-2019.

Krajewski, W.F., Ceynar, D., Demir, I., Goska, R., Kruger, A., Langel, C., Mantilla, R., Niemeier, J., Quintero, F., Seo, B.-C., Small, S.J., Weber, L.J., Young, N.C., 2017. Real-time flood forecasting and information system for the state of iowa. Bull. Am. Meteorol. Soc. 98, 539–554. https://doi.org/10.1175/BAMS-D-15-00243.1.

Krajewski, W.F., Ghimire, G.R., Quintero, F., 2020. Streamflow Forecasting without Models. J. Hydrometeorol. 21, 1689–1704. https://doi.org/10.1175/JHM-D-19-0292.1.

Krajewski, W.F., Ghimire, G.R., Demir, I., Mantilla, R., 2021. Real-time streamflow forecasting: AI vs. Hydrologic Insights. J Hydrol X 13, 100110. https://doi.org/10.1016/j.hydroa.2021.100110.

Krajewski, W.F., Smith, J.A., 2002. Radar hydrology: rainfall estimation. Adv. Water Resour. 25, 1387–1394. https://doi.org/10.1016/S0309-1708(02)00062-3.

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, 6005–6022. https://doi.org/10.5194/hess-22-6005-2018.

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A.K., Hochreiter, S., Nearing, G.S., 2019a. Toward improved predictions in ungauged basins: exploiting the power of machine learning. Water Resour. Res. 55, 11344–11354. https://doi.org/10.1029/2019WR026065.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., Nearing, G., 2019b. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. Hydrol. Earth Syst. Sci. 23, 5089–5110. https://doi.org/10.5194/HESS-23-5089-2019.

Kumar, D.N., Srinivasa Raju, K., Sathish, T., 2004. River flow forecasting using recurrent neural networks. Water Resour. Manag. 18, 143–161. https://doi.org/10.1023/B:WARM.0000024727.94701.12.

Lamontagne, J.R., Barber, C.A., Vogel, R.M., 2020. Improved estimators of model performance efficiency for skewed hydrologic data. Water Resour. Res. 56, e2020WR027101 https://doi.org/10.1029/2020WR027101.

Mai, J., Shen, H., Tolson, B.A., Gaborit, É., Arsenault, R., Craig, J.R., Fortin, V., Fry, L.M., Gauch, M., Klotz, D., Kratzert, F., O'Brien, N., Princz, D.G., Rasiya Koya, S., Roy, T., Seglenieks, F., Shrestha, N.K., Temgoua, A.G.T., Vionnet, V., Waddell, J.W., 2022. The great lakes runoff intercomparison project phase 4: the great lakes (GRIP-GL). Hydrol. Earth Syst. Sci. 26, 3537–3572. https://doi.org/10.5194/hess-26-3537-2022.

Mantilla, R., 2007. Physical Basis of Statistical Scaling in Peak Flows and Stream Flow Hydrographs for Topologic and Spatially Embedded Random Self-similar Channel Networks. University of Colorado.

Mantilla, R., Gupta, V.K., 2005. A GIS numerical framework to study the process basis of scaling statistics in river networks. IEEE Geosci. Remote Sens. Lett. 2, 404–408. https://doi.org/10.1109/LGRS.2005.853571.

Mantilla, R., Gupta, V.K., Mesa, J.O., 2006. Role of coupled flow dynamics and real network structures on Hortonian scaling of peak flows. J Hydrol (amst) 322, 155–167. https://doi.org/10.1016/j.jhydrol.2005.03.022.

Mantilla, R., Krajewski, W.F., Velásquez, N., Small, S.J., Ayalew, T.B., Quintero, F., Jadidoleslam, N., Fonley, M., 2022. The hydrological Hillslope-Link Model for space-time prediction of streamflow: Insights and applications at the Iowa Flood Center. In: Marina, A., Nikolopoulos, E. (Eds.), Extreme Weather Forecasting. Elsevier.

Mantillla, R., Perez, G., Velasquez, N., Wright, D.B., Yu, G., 2021. Regional flood frequency analysis using physics-based hydrologic modeling. ESS Open Archive. https://doi.org/10.1002/essoar.10506017.1.

Muhammad, A.U., Li, X., Feng, J., 2019. Using LSTM GRU and Hybrid Models for Streamflow Forecasting. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST 294 LNCIST, 510–524. https://doi.org/10.1007/978-3-030-32388-2_44/COVER.

Oyebode, O., Stretch, D., 2019. Neural network modeling of hydrological systems: a review of implementation techniques. Nat. Resour. Model. 32, e12189.

Perez, G., Mantilla, R., Krajewski, W.F., Wright, D.B., 2019. Using physically based synthetic peak flows to assess local and regional flood frequency analysis methods. Water Resour. Res. 55, 8384–8403. https://doi.org/10.1029/2019WR024827.

Quintero, F., Krajewski, W.F., Seo, B.-C., Mantilla, R., 2020. Improvement and evaluation of the iowa flood center hillslope link model (HLM) by calibration-free approach. J Hydrol (amst) 584, 124686. https://doi.org/10.1016/j.jhydrol.2020.124686.

Ravanelli, M., Brakel, P., Omologo, M., Bengio, Y., 2018. Light gated recurrent units for speech recognition. IEEE Trans Emerg Top Comput Intell 2, 92–102. https://doi.org/10.1109/TETCI.2017.2762739.

Sharma, S., Raj Ghimire, G., Siddique, R., 2023. Machine learning for postprocessing ensemble streamflow forecasts. J. Hydroinf. 25, 126–139. https://doi.org/10.2166/hydro.2022.114.

Small, S.J., Jay, L.O., Mantilla, R., Curtu, R., Cunha, L.K., Fonley, M., Krajewski, W.F., 2013. An asynchronous solver for systems of ODEs linked by a directed tree structure. Adv. Water Resour. 53, 23–32. https://doi.org/10.1016/j.advwatres.2012.10.011.

Sutskever, I., Vinyals, O., Le, Q. V., 2014. Sequence to Sequence Learning with Neural Networks. NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems 3104–3112.

Velásquez, N., Mantilla, R., Krajewski, W., Fonley, M., Quintero, F., 2021. Improving hillslope link model performance from non-linear representation of natural and artificially drained subsurface flows. Hydrology 8, 187. https://doi.org/10.3390/hydrology8040187.

Vidyarthi, V.K., Jain, A., Chourasiya, S., 2020. Modeling rainfall-runoff process using artificial neural network with emphasis on parameter sensitivity. Model Earth Syst Environ 6, 2177–2188. https://doi.org/10.1007/s40808-020-00833-7.

Wan, X., Yang, Q., Jiang, P., Zhong, P., 2019. A hybrid model for real-time probabilistic flood forecasting using elman neural network with heterogeneity of error distributions. Water Resour. Manag. 33, 4027–4050. https://doi.org/10.1007/s11269-019-02351-3.

Wang, S., Hu, Y., Burgues, J., Marco, S., Liu, S.-C., 2020. Prediction of Gas Concentration Using Gated Recurrent Neural Networks, in: 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS). IEEE, pp. 178–182. https://doi.org/10.1109/AICAS48895.2020.9073806.

Wright, D.B., Yu, G., England, J.F., 2020. Six decades of rainfall and flood frequency analysis using stochastic storm transposition: Review, progress, and prospects. https://doi.org/10.1016/j.jhydrol.2020.124816.

Wright, D.B., Smith, J.A., Villarini, G., Baeck, M.L., 2013. Estimating the frequency of extreme rainfall using weather radar and stochastic storm transposition. J Hydrol (amst) 488, 150–165. https://doi.org/10.1016/j.jhydrol.2013.03.003.

Wright, D.B., Mantilla, R., Peters-Lidard, C.D., 2017. A remote sensing-based tool for assessing rainfall-driven hazards. Environ. Model. Softw. 90, 34–54. https://doi.org/10.1016/j.envsoft.2016.12.006.

Xiang, Z., Demir, I., Mantilla, R., Krajewski, W.F., 2021. A Regional Semi-Distributed Streamflow Model Using Deep Learning. https://doi.org/https://doi.org/10.31223/X5GW3V.

Xiang, Z., Yan, J., Demir, I., 2020. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resour. Res. 56, e2019WR025326 https://doi.org/10.1029/2019WR025326.

Yu, G., Wright, D.B., Zhu, Z., Smith, C., Holman, K.D., 2019. Process-based flood frequency analysis in an agricultural watershed exhibiting nonstationary flood seasonality. Hydrol. Earth Syst. Sci. 23, 2225–2243. https://doi.org/10.5194/HESS-23-2225-2019.

Zhu, Z., Wright, D.B., Yu, G., 2018. The impact of rainfall space-time structure in flood frequency analysis. Water Resour. Res. 54, 8983–8998. https://doi.org/10.1029/2018WR023550.