JGR | Machine Learning and Computation

RESEARCH ARTICLE
10.1029/2024JH000405

A Proof of Concept for Improving Estimates of Ungauged Basin Streamflow via an LSTM-Based Synthetic Network Simulation Approach

A. A. Ramírez Molina[1], J. M. Frame[2], J. Halgren[3], and J. Gong[1]

[1]Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA, [2]Department of Geological Sciences, The University of Alabama, Tuscaloosa, AL, USA, [3]Alabama Water Institute, The University of Alabama, Tuscaloosa, AL, USA

Key Points:

- The study uses a deep learning model to present an innovative method for improving historic river flow estimates in unmonitored areas
- The proof of concept uses synthetic hydrographs from simplified basin hydrology bucket models that split flow into baseflow and runoff
- The synthetic upstream-downstream basin pairs mirror ungauged basins in a monitored basin, allowing the concept to be extended to real data

Correspondence to:

A. A. Ramírez Molina and J. Gong,
aaramirez@crimson.ua.edu;
jiaqi.gong@ua.edu

Citation:

Ramírez Molina, A. A., Frame, J. M., Halgren, J., & Gong, J. (2025). A proof of concept for improving estimates of ungauged basin streamflow via an LSTM-based synthetic network simulation approach. *Journal of Geophysical Research: Machine Learning and Computation*, *2*, e2024JH000405. https://doi.org/10.1029/2024JH000405

Received 22 AUG 2024
Accepted 19 APR 2025

Author Contributions:

Conceptualization: A. A. Ramírez Molina, J. M. Frame, J. Halgren, J. Gong
Data curation: A. A. Ramírez Molina
Formal analysis: A. A. Ramírez Molina, J. Halgren
Funding acquisition: J. Halgren
Investigation: A. A. Ramírez Molina, J. Halgren
Methodology: A. A. Ramírez Molina, J. M. Frame, J. Halgren
Project administration: A. A. Ramírez Molina, J. Halgren
Resources: J. Halgren
Software: A. A. Ramírez Molina, J. M. Frame

Abstract This study introduces a machine learning approach to address the critical challenge of limited real-time flow data in river basins, particularly for calibrating large-scale hydrologic models. These models often rely on uncertain parameter transfers from gauged to ungauged regions, hindering accurate predictions. To mitigate this, we propose utilizing long short-term memory (LSTM) networks to estimate historic streamflow in ungauged watersheds, effectively creating "surrogate gauges." Our innovative method treats watersheds as interconnected systems, leveraging downstream flow information to constrain and improve upstream flow estimates. By training and testing the LSTM on synthetic networks of conceptual "leaky bucket" watershed models, we demonstrate that it outperforms traditional methods. Notably, by representing watersheds as paired upstream-downstream networks, this concept can be generalized to any ungauged portion of a real basin, enhancing flow estimates in data-scarce scenarios. This research represents a proof of concept, highlighting the potential for significantly enhancing hydrologic modeling in data-scarce regions and providing a scalable method for watershed network analysis. Future work will focus on applying the model to real-world basins to validate its performance and scalability for continental-scale applications. Ultimately, this research aims to contribute to the development of more accurate and reliable hydrologic predictions.

Plain Language Summary This study presents a machine learning method to address the challenge of limited real-time flow data in river basins, especially for improving the accuracy of large-scale water flow predictions. These models often use uncertain data from monitored areas to make predictions for unmonitored regions, which can lead to inaccuracies. To mitigate this, we propose using a Long-Short-Term Memory (LSTM) model, which mimics human memory, to estimate past water flow in unmonitored areas, effectively creating "surrogate gauges." Our method treats watersheds as interconnected systems, leveraging known downstream flow information to improve estimates of unknown upstream flow. By training and testing the LSTM on synthetic networks that simulate real watersheds using "leaky bucket" models, we demonstrate its superiority over traditional methods. Importantly, by modeling watersheds as connected upstream-downstream pairs, this approach can be applied to any unmonitored segment of a real basin, improving flow estimates where data are scarce. This research demonstrates a promising idea, showing the potential to greatly improve water flow modeling in areas with little data, and offering a scalable way to analyze watershed networks. Future work will involve using this model on actual river basins to confirm its effectiveness and ability to scale up for larger, continental-wide applications.

1. Introduction

Effective management of water resources is a major challenge, especially in mitigating natural events like floods and droughts. Data from stream gauges, which measure the flow in river channels, are a critical resource for preparing water resource models; however, many places do not have enough gauge data to support accurate modeling. For instance, in the United States, it is notable that less than 10% of the river segments are actively monitored using gauges, which leads to a gap in the data sets necessary to predict water-related events such as floods and droughts, particularly on local scales (National Oceanic and Atmospheric Administration (NOAA), 2016). The National Water Model (NWM) of the NOAA covers about 2.7 million stream segments over 3.2 million miles in the United States, representing a significant advancement in water modeling and analysis. However, calibration and evaluation of this model is limited by the lack of real-time streamflow monitoring data

Supervision: J. M. Frame, J. Halgren, J. Gong
Validation: A. A. Ramírez Molina, J. Halgren
Visualization: A. A. Ramírez Molina
Writing – original draft: A. A. Ramírez Molina
Writing – review & editing: A. A. Ramírez Molina, J. Halgren, J. Gong

for more than 90% of these segments, which restricts the ability to make accurate and timely predictions of water-related events (Cosgrove et al., 2024).

Traditionally, streamflow in unmonitored river segments is estimated using the physical and meteorological characteristics of individual catchment areas combined using methods similar to those developed to support streamflow forecasting. The use of forecast techniques is appropriate, given that the problem is similar—in forecasting, we are using calibrated parameters to simulate basin behavior in an unobserved time; in the case of estimating historic flows in an ungauged basin, we are using known patterns of behavior transferred to an unobserved location. Building on the findings of Kratzert et al. (2019), which demonstrated the effectiveness of using static watershed characteristics and meteorological data to predict flow in ungauged basins, this study aims to extend these insights into a more complex hydrological context. Part of the difficulty with forecast methods is that they must assume a certain lack of information that, while proper for prognostic modeling, ignores the additional information available from existing flow measurements when applying such methods for estimation of historic flows in ungauged basins.

In future predictions, confidence in related flows across different points may vary, especially when moving from in-sample to out-sample testing. However, for historic flows, any downstream observations are particularly valuable for constraining estimates at the unmonitored location. The downstream gauge measurement will retain a certain amount of the signal from the upstream location and we will show that by incorporating the information in the downstream measurement in a deep learning model, we can obtain a more accurate estimate of upstream ungauged flows than those obtained through traditional forecast methods alone.

The inspiration for this work comes in part from the successful application of deep learning models for signal analysis in other domains. In particular, Ephrat et al. (2018) showed that deep learning methods may be used for audio signal separation in noisy environments, which is analogous in some ways to the separation of ungauged tributary flow signals from the combined signal of a downstream gauge. Long short-term memory (LSTM) models, as developed by Hochreiter and Schmidhuber (1997), are recognized for their ability to analyze patterns in temporal data and have been shown to be ideally suited for modeling river flow dynamics (e.g., Kratzert et al. (2019); Xu et al. (2020)). As early as Krauße (2007), LSTMs have been identified as particularly promising for hydrological modeling because their internal state can represent both short-term and long-term hydrologic processes, essential for capturing watershed states and rainfall-runoff dynamics. LSTM are the deep learning method applied in this study.

Building on this recognition of LSTM's suitability for hydrologic modeling, the current study leverages synthetic data experiments to further advance theoretical applications of LSTM in hydrology. Specifically, an innovative methodology using an LSTM network is proposed for the generation of synthetic river flow data. Based on the "leaky bucket" model, a conceptual approach widely applied in hydrology to simplify the representation of water storage and release in natural systems. The model likens a watershed or river segment to a bucket that collects water (e.g., from precipitation) and releases it through a spigot (representing runoff or deep infiltration). This simplified structure allows for modeling key hydrological processes such as evapotranspiration, infiltration, and precipitation in a computationally efficient way (National Oceanic and Atmospheric Administration (NOAA), 2023). Networks of these "buckets" are developed to simulate watersheds, where multiple tributary watershed "buckets" converge into a main river, laying the foundation for synthetic data generation.

Training a deep learning model to estimate upstream flows based on downstream observations would ideally rely on extensive data sets containing simultaneous flow measurements at both downstream and upstream locations. However, such fully monitored networks are rare, limiting the practicality of directly applying this approach with real-world data. To clearly demonstrate the feasibility of our proposed methodology as a proof of concept, we first employ synthetic data sets generated by simplified hydrologic models (Leaky Bucket networks). These synthetic scenarios offer numerous fully monitored examples, enabling rigorous testing and validation of the concept before extending its applicability to real-world situations.

The purpose of this paper is to introduce a novel approach for estimating historical flows in ungauged basins by leveraging downstream gauge observations as additional constraints. Unlike traditional methods that rely solely on individual basin characteristics, our approach demonstrates that incorporating downstream flow information significantly enhances prediction accuracy of upstream ungauged flows. We present a novel conceptualization using pairs of connected basins (one gauged, one ungauged) as a generalizable framework applicable to any

upstream-downstream relationship within a river network. Using synthetic data generated from bucket models as a proof of concept, we show that this paired approach, implemented through LSTM networks, produces more accurate flow estimates than models trained on individual basins alone. While this initial demonstration uses synthetic data, the methodology establishes a foundation for future applications with real gauge data, with the potential to provide high-confidence surrogate historical flow records that could serve as validation benchmarks for predictive models in ungauged basins.

2. Related Work

This study builds on recent advances in computational hydrology and deep learning, with a specific focus on two core concepts: the "Leaky Bucket" approach to conceptual watershed modeling, and the use of LSTM networks to capture complex temporal dependencies. In this section, four areas of associated research are examined that form the foundation of the proposed approach.

2.1. Leaky Bucket Models for Streamflow

A seminal idea in hydrological modeling was introduced by Nash (1957), who viewed watersheds as a series of linear reservoirs (or "buckets"). More recently, various "leaky bucket" formulations have provided a single-container analog, into which precipitation enters and from which water is released via baseflow, overflow, and other fluxes. Although simple, these models have proven effective in capturing essential hydrologic processes, such as the partitioning of total streamflow into baseflow (via spigot flow) and stormflow (via overflow), especially in data-scarce settings (National Oceanic and Atmospheric Administration (NOAA), 2023; Terink et al., 2015). The simplified representation employed in this study intentionally prioritizes computational efficiency and theoretical clarity, focusing specifically on these key flow regimes (baseflow and stormflow) to evaluate the proposed methodological contribution.

Hybrid approaches embed similar storage-release concepts within neural networks. For example, Jiang et al. (2020) and Feng et al. (2023) combine physically inspired reservoir principles with Recurrent Neural Network (RNNs) architectures to improve extrapolation in ungauged regions. Similarly, Wang and Gupta (2024) introduce a mass-conserving perceptron to enforce physical constraints within neural networks, illustrating the broader trend of pairing conceptual hydrologic elements with machine learning.

Tools like the "DeepBucketLab" (Frame et al., 2024) further simplify this paradigm for teaching and experimentation, generating synthetic data with bucket models and training neural networks to predict flows. While real basins are often more complex, such simplified conceptualizations are a practical way to test how downstream observations can help constrain upstream flow estimates.

2.2. Streamflow Prediction in Ungauged Basins

A long-standing challenge in hydrology is estimating flow in ungauged basins, which lack direct flow measurements. Traditional methods often rely on the physical and meteorological characteristics of a catchment, transferring model parameters from nearby "donor" watersheds (Adams, 2021; Vrugt et al., 2006). While these methods are widely accepted—particularly in data-scarce regions—they can struggle to represent highly nonlinear processes or the interactions among multiple segments in a basin (Hrachowitz et al., 2013; Sivapalan et al., 2003). As part of the international Predictions in Ungauged Basins initiative, numerous studies strove to improve hydrological predictions by combining watershed attributes and historical climate data, underscoring ongoing challenges in achieving high accuracy without real-time observations (Hrachowitz et al., 2013; Sivapalan et al., 2003).

More recent research leverages machine learning to tackle these limitations. For instance, Kratzert et al. (2019) showed that LSTM networks can capture complex, non-linear relationships between meteorological forcings and streamflow, outperforming traditional models in many ungauged settings. Likewise, Wilbrand et al. (2023) developed LSTM models for over 500 basins, finding that although local data sets often yield better results, global-scale models remain competitive in certain regions. Other efforts, such as the Inverse Streamflow Routing model (Fisher et al., 2020), propagate discharge data through a river network using a Kalman smoother, while Perera et al. (2024) employ hybrid methods with explainable AI for extreme events. Despite these advances, many

approaches emphasize parameter interpolation or purely meteorological inputs, rather than harnessing combined downstream flows to constrain upstream estimates.

The present study specifically addresses this gap by exploring how historic downstream flow measurements can serve as a strong "mixed-signal" constraint, improving estimates of individual upstream tributaries via an LSTM-based surrogate modeling framework. This approach builds on the premise that gauged downstream records—often available retrospectively—hold valuable information about ungauged upstream segments.

2.3. Deep Learning Signal Separation

A useful analogy for the upstream–downstream problem can be found in audio processing. Ephrat et al. (2018) proposed a deep learning model that separates multiple speakers' voices from a single, noisy audio mixture. They combined two data streams—visual cues (e.g., lip movements) extracted by a convolutional neural network, and the raw audio waveform—and then passed these features to an LSTM to disentangle individual voice signals. By jointly analyzing "who is speaking" (via the CNN's visual features) and "what is being said" (via the LSTM's audio processing), the system could isolate each speaker more accurately than audio-only methods.

In the hydrologic context, the downstream flow is analogous to a "mixed signal," containing the combined contributions from multiple upstream tributaries. The physical and meteorological variables (e.g., infiltration, precipitation, and evapotranspiration) are akin to Ephrat et al. (2018)'s visual features, providing additional clues that help the LSTM "separate" each tributary's flow within the aggregated downstream record. This cross-domain analogy illustrates how deep learning can parse a complex, composite signal into distinct components, highlighting the versatility and power of LSTM-based architectures for both audio and hydrologic signal separation.

2.4. LSTM Simulation of Watershed Processes

LSTM networks (Hochreiter & Schmidhuber, 1997) are recognized for effectively modeling time-series data, owing to their ability to retain and update hidden states over multiple timesteps. In hydrology, studies have shown that LSTM networks can outperform traditional models by capturing nonlinear rainfall-runoff relationships more accurately. For example, Xu et al. (2020) applied LSTMs to both daily and 10-day flow predictions in two Chinese basins, demonstrating superior performance compared to alternative machine learning methods. Hyperparameter tuning was found to be crucial for maximizing efficiency, particularly regarding the inclusion or placement of fully connected layers and the selection of sequence length.

Other research has extended LSTM applications to large-scale regional studies. Kratzert et al. (2019) used LSTM networks and static basin attributes to predict streamflow across hundreds of basins in the continental United States, achieving median Nash–Sutcliffe efficiencies of 0.69 for out-of-sample tests. Such results highlight the adaptability of LSTMs to diverse geographical and climatic conditions, including data-scarce regions. Likewise, Hu et al. (2020) showed that LSTMs can handle rapid-onset flood events in small rivers, capturing peak flows more reliably than classical models or simpler machine learning approaches. Improvements in metrics such as root mean square error (RMSE) and mean absolute error (MAE) have also been reported when using additional meteorological variables as inputs (Fan et al., 2020).

LSTM networks have thus proven effective for a broad range of hydrologic tasks, including flood forecasting, low-flow prediction, and runoff simulation. Although other machine learning methods (such as artificial neural networks or support vector regression) can produce comparable results in certain settings, LSTMs are specifically designed to handle time-varying signals with complex temporal dependencies. This makes them especially suitable for capturing fluctuations in streamflow, whether at regional or local scales. Additionally, tools like the "DeepBucketLab" (Frame et al., 2024) illustrate how synthetic LSTM experiments can be used to train and evaluate hydrologic models, particularly those involving simplified bucket frameworks. Such controlled setups are valuable for testing novel ideas before translating them to real-world data.

2.5. Potential Applications and Scope

While this study focuses on a proof of concept using synthetic data, the approach of leveraging downstream observations to constrain upstream flow estimates has potential applications in operational frameworks like NOAA's NWM (National Oceanic and Atmospheric Administration (NOAA), 2016). The NWM simulates millions of river segments across the United States, many of which remain unmonitored (U.S. Geological

Survey, 2024). Methodologies that can extract information from available gauged locations to improve estimates in ungauged tributaries could complement existing high-resolution modeling efforts, though such integration lies beyond this paper's scope.

3. Methodology

This study focuses on the use of combined downstream flow data from networks of converging tributaries along with meteorological data to predict individual upstream flows. Recognizing the converging structure of watershed networks—where multiple tributaries create a singular downstream flow that encapsulates the integrated signals from upstream processes—this approach aims to accurately disaggregate these flows back into their originating sources using LSTM models.

To achieve this, large and detailed data sets are necessary to effectively train and validate the models. But with the scarcity of comprehensive real-world hydrological data from fully gauged networks, synthetic data sets are created to develop the proof of concept evaluated in this study. These synthetic data sets simulate the extensive monitoring required for detailed watershed network analysis, providing a robust training environment that mimics the variability and complexity of real-world scenarios.

The methodology used in this study builds upon the "DeepBucketLab" framework developed by Frame et al. (2024), which employs a "leaky bucket" model for individual river segments and integrates an LSTM network for predicting spigot flow and overflow. We build upon this approach by creating a network of interconnected buckets and training the LSTM model using both individual and combined bucket data.

To implement the concept of developing the synthetic stream gauges central to this study, a Leaky Bucket model was initially employed to simulate individual river segments, allowing for each to have unique physical and meteorological characteristics. These models were then systematically organized into networks that represent entire watersheds. Within these networks, each tributary is simulated by an individual "bucket," and their respective streamflows are integrated to form a singular "downstream flow." For the purpose of training, validating, and testing, data sets composed of these bucket networks were constructed. These data sets were subsequently utilized to develop a LSTM model, designed to disaggregate the combined "downstream flow" back into the individual "upstream flows" of each tributary within the network.

3.1. Leaky Bucket Model

As defined earlier, the "Leaky Bucket" model simulates the storage and release of water in systems such as watersheds, soils or even in climate models (MacBean et al., 2020; National Oceanic and Atmospheric Administration (NOAA), 2023; Terink et al., 2015). The model is conceptualized as a bucket that is filled with water (e.g., from rainfall) and has a spigot through which the water escapes or leaks (e.g., as runoff or deep infiltration). This simplified idea helps to understand and model the behavior of water in a natural or artificial system.

The "Leaky Bucket" model implemented in this study uses a conceptual modeling approach that simulates a river segment, taking into account its physical and climatological characteristics. Figure 1 shows a schematic of the basic model developed to represent in a simplified way hydrological processes in a river such as precipitation, evapotranspiration, infiltration, overflow and flow through the spigot.

3.1.1. Model Characteristics and Parameters

In the implemented model, each bucket simulates a specific segment of a river and is characterized by several key parameters. These include the bucket area (A_{bucket}), which determines the maximum water storage capacity; the spigot area (A_{spigot}), which influences the outflow rate; the bucket height (H_{bucket}), which defines the threshold for overflow; the water height (h_{bucket}), which indicates the current amount of water in storage and directly affects the water pressure in the spigot; the height of the spigot (H_{spigot}), which affects the flow through the spigot; the distance from the spigot to the water level (h_{spigot}), which regulates the hydrostatic pressure in the spigot and thus impacts the water outflow velocity; the infiltration coefficient ($K_{\text{infiltration}}$), which represents the water loss by infiltration to the soil; and an evapotranspiration parameter ($ET_{\text{parameter}}$), which modulates the water loss by

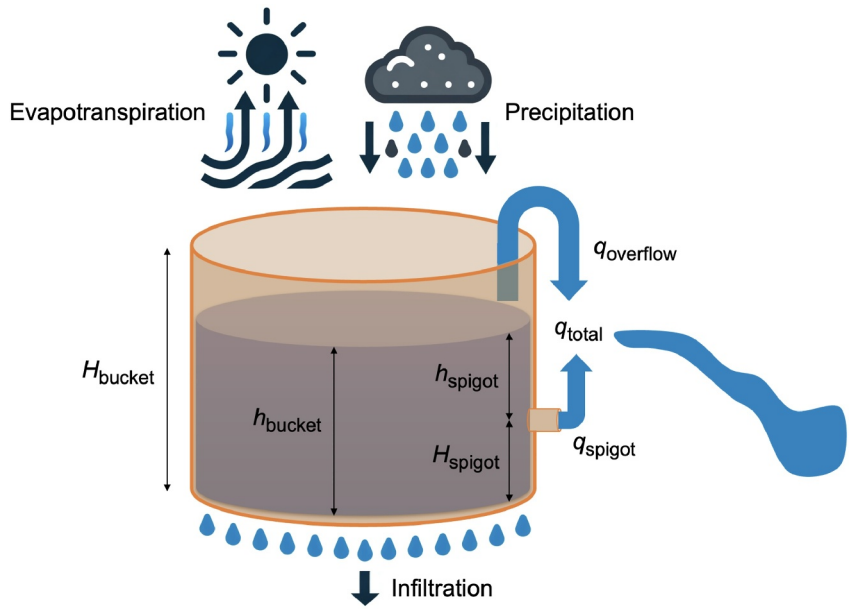


Figure 1. Diagram of implemented Leaky Bucket model.

evaporation and transpiration. Each of these parameters is randomly generated within predefined ranges to reflect the natural diversity of watershed characteristics.

3.1.2. Model Dynamics

The model operates in discrete time steps, during which it simulates the following hydrological processes:

1. *Precipitation*: Represented as an input of water into the bucket, the amount of precipitation varies randomly, but the probability of rainfall on a given day is influenced by the rainfall of the previous day. This approach ensures that transitions between different types of rainfall (none, light, or heavy) are realistic, avoiding abrupt shifts between heavy rain and completely dry days. This process increases the water level in the bucket.
2. *Evapotranspiration (ET)*: The ET water loss is calculated as a function of the area of the bucket, modulated by a sinusoidal function of time and the ET parameter of the bucket, including a noise factor introduced to simulate random variations in ET, as shown in Equation 1. This process reduces the water level.

$$ET = \max\left(0, \frac{A_{\text{bucket}}}{ET_{\text{parameter}}} \cdot \sin(t) \cdot \text{noise}(t)\right) \tag{1}$$

3. *Infiltration*: Modeled as a proportional reduction of the water level in the bucket per unit time, as a function of the current depth in the bucket and the infiltration coefficient Equation 2, symbolizing the percolation of water into the underlying soil.

$$\text{Infiltration} = h_{\text{bucket}} \times K_{\text{infiltration}} \tag{2}$$

4. *Overflow*: Occurs when the water level exceeds the height of the bucket, representing situations of excess water such as during heavy rains or floods.
5. *Outflow through the Spigot (Contributions to stream network)*: Based on the principles of fluid mechanics, the velocity of outflow through the spigot is initially calculated using the Torricelli equation Equation 3, where the velocity depends on the height of the water above the spigot and the acceleration due to gravity (Gerhart et al., 2021). To determine the total volume of outflow (q_{spigot}), this velocity is then multiplied by the area of

the spigot and the time step (Δt), as shown in Equation 4. This flow simulates the natural discharge of water from the river segment.

$$v = \sqrt{2gh} \tag{3}$$

$$q_{\text{spigot}} = v \times A_{\text{spigot}} \times \Delta t \tag{4}$$

Additional illustrations of the simulated inputs and outputs for the Leaky Bucket model (e.g., precipitation, evapotranspiration, and flow through the spigot) are provided in Appendix A, Section A1.

3.1.3. Integration of Processes in the Model

Each of these processes interacts within the model to dynamically reflect the hydrologic behavior of the represented river segment. Precipitation increases the water level, while evaporation, transpiration, and infiltration decrease it. Overflow and flow through the spigot are the output mechanisms that modulate the system's response to water inputs, thus simulating the storage and release of water in river segments.

3.2. Bucket Networks

In the following, the representation of hydrographic basins and their hydrodynamic simulation through networks of basins is discussed. The structure of watersheds is introduced as directed acyclic graphs to simulate the streamflow from tributaries to main rivers. Furthermore, it is examined how these streamflows combine to fully reflect the dynamics of the watershed.

The networks are designed to integrate the flows of several river segments, simulating the interconnection and dynamics of a real watershed. This study intentionally employs synthetic data to provide a controlled environment for testing the core concept of using combined downstream flows to improve upstream predictions. This approach enables an isolated evaluation of the methodology, minimizing confounding factors inherent in real-world data.

3.2.1. Watershed Representation

A watershed can be represented as a directed acyclic graph (DAG), since each tributary of a river flows in a single direction, from upstream to downstream, and there are no closed cycles in it. Figure 2 shows the DAG representation of a watershed made up of two river segments that are individually represented by Leaky Bucket Models.

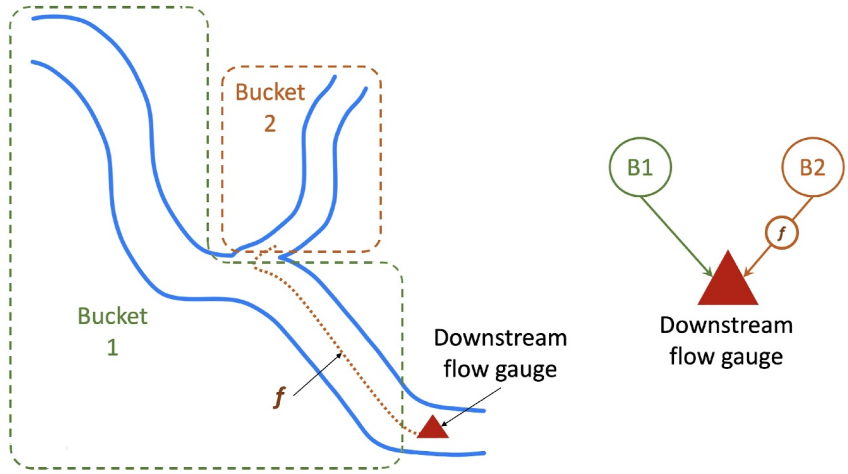
In this study, a DAG is referred to as a Bucket Network. Thus, a bucket network represents a complete watershed, where each bucket simulates a single tributary segment. The networks are designed to integrate the flows of several river segments, simulating the interconnection and dynamics of a real watershed.

The network configuration of the buckets allows the model to simulate not only water flows in individual segments, but also their interaction throughout the entire basin. Representing watersheds with networks of two buckets allows for generalization to any ungauged portion of a basin. This is because any watershed with a downstream gauge can be modeled with one bucket representing the monitored downstream segment and another bucket representing any of its ungauged tributaries.

Within each network, the outflows from each bucket (simulating tributary segments) combine to form a 'downstream' streamflow. This combined flow represents the sum total of the water flowing through the network, similar to how water flows are combined in a real watershed. This configuration enables the simulation of upstream–downstream interactions between tributaries, offering a more structured representation of watershed connectivity than models that treat each river segment in isolation.

3.2.2. Combined Downstream Flow

In order to satisfactorily represent a watershed with a bucket network, it is necessary to combine the outflows of each bucket Q_i within the network into a single one Q_{Total} using a flow routing function f. As shown in Equation 5.



(a) River representation (b) DAG representation

Figure 2. Representation of a watershed as a network of tributaries that converge in a single river whose flow is observed by the downstream gauge. *f* represents the routing function that conveys the flow from the tributary bucket (Bucket 2) to the downstream gauge. (a) Shows how each river segment can be represented by a Leaky Bucket Model. (b) Shows the representation of this network as a directed acyclic graph.

$$Q_{\text{Total}} = \sum_{i=1}^{n} f(Q_i) \tag{5}$$

Further details on how individual bucket flows are integrated into a single downstream flow, along with the construction of training, validation, and test data sets, can be found in Appendix A, Section A2.

For the purposes of this proof of concept, the upstream flows from each bucket were combined using a null routing function. This simplified approach allowed us to focus on the model's ability to separate the downstream flow into the respective upstream contributions. Future research could explore more complex routing methods for a more accurate representation of flow dynamics.

3.3. Metrics

The effectiveness of the model was evaluated by applying three metrics: RMSE, MAE and Nash-Sutcliffe Efficiency (NSE). These metrics were calculated using the validation data sets to ensure robustness and minimize overfitting.

The RMSE is a common statistic in regression analysis that measures the average size of the errors between predicted and observed values. In situations where significant errors must be avoided, the RMSE is a valuable tool. It exhibits a high degree of sensitivity to outliers and assigns greater significance to larger errors (Hyndman & Koehler, 2006). It is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

where y_i are the observed values and $\hat{y}_i$ are the predicted values.

The MAE is a linear score that measures the average magnitude of the errors in a set of predictions, regardless of their direction. The MAE is not impacted by outliers, which makes it useful for regression tasks with data sets where outliers exist (Hyndman & Koehler, 2006). It is calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

The NSE proposed by Nash and Sutcliffe in 1970, is frequently used in hydrology to evaluate the predictive accuracy of hydrological models (Nash & Sutcliffe, 1970). The normalized outcome of NSE ranges from $-\infty$ to 1, making it easier to compare across various models. Nonetheless, NSE fails to yield values in scenarios where the observed variance is zero, and it is notably sensitive to extreme values (Ramírez Molina et al., 2023). It is defined as:

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \tag{8}$$

where $\bar{y}$ is the mean of the observed values.

These metrics provide a comprehensive view of the model's performance, allowing to evaluate its accuracy and robustness in predicting hydrological time series.

3.4. LSTM Model

Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) were chosen due to their aptitude for sequence data and ability to address vanishing/exploding gradient issues in traditional RNNs (Goodfellow et al., 2016). By employing memory units, often called LSTM cells, these architectures use gates to regulate information flow, allowing them to retain long-range temporal dependencies. In the present study, LSTMs are used to capture upstream flow signals from the combined downstream time series. Each input sequence comprises the simulated physical and meteorological features of each bucket (e.g., precipitation, infiltration, evapotranspiration), along with the network's combined downstream flow, and the model subsequently outputs separate upstream flows per bucket. This approach leverages LSTMs' inherent suitability for sequential data modeling in hydrology. Complete gate equations, the final network architecture, and the hyperparameter optimization procedure are provided in Appendix B.

4. Results

This section presents the results obtained from applying the proposed methodology across multiple experiments. First, the performance of the optimized LSTM model is examined on the test data set, confirming its robustness beyond the validation period. Next, a comparative analysis demonstrates how including downstream flow observations ("Combined bucket network approach") outperforms training solely on individual bucket data ("Individual bucket approach"). A subsequent experiment then explores the model's behavior under low-flow and high-flow extremes, providing further insight into its performance across diverse hydrologic regimes. Together, these results highlight the potential benefits of leveraging downstream measurements to improve upstream flow estimates in ungauged scenarios under a wide range of conditions.

4.1. Model Evaluation and Performance

The optimized model, identified via hyperparameter optimization with the validation data set, was applied to make predictions on the test data set. The performance results obtained are summarized in Table 1.

The visual comparison of the predicted values against the "actual" outputs from the Leaky Bucket model, as illustrated in Figure 3, demonstrates a close correspondence between the predicted and true values. The results obtained from the test data set further confirm the findings from the validation data set, demonstrating the robustness and generalizability of the optimized model.

4.2. Comparison of Flow Estimation Methods

To further validate the proposed model, a comparative analysis between two approaches was performed using the same synthetic test data set. In the first approach, hereafter referred to as the "Individual bucket approach," estimates of the spigot flow and overflow of each upstream bucket were made by training the LSTM model only with the individual physical and meteorological characteristics of each bucket, without incorporating the combined downstream flow or the characteristics of other buckets. The second approach, which is the method proposed in the present study and is referred to as the "Combined bucket network approach," involves training the

Table 1
Performance Metrics for the Optimized Model on Test Data Set

	Spigot flow			Overflow		
Model	RMSE	MAE	NSE	RMSE	MAE	NSE
Optimized Model	0.144	0.109	0.886	0.210	0.087	0.961

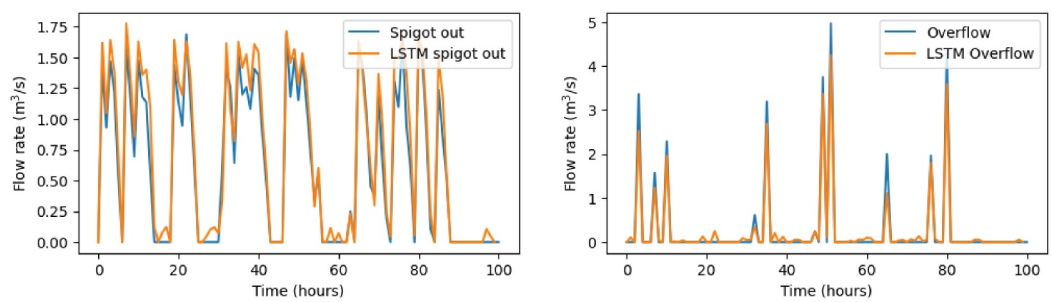
LSTM model using both the physical and meteorological characteristics of all buckets in the network along with the information from the "observed" combined downstream flow.

The comparison results shown in Table 2 clearly indicate that incorporating downstream flow data significantly enhances the model's ability to predict spigot flow and overflow in ungauged portions of a synthetic basin. In other words, the model can effectively "separate" upstream flow signals from the combined downstream flow. To further examine how widespread this improvement was, a bucket-level comparison was conducted across the 204 networks in the test data set (408 buckets total). The results show that the combined approach yielded better performance metrics for 89% of all buckets, including 326 cases (approximately 80%) with improved spigot flow predictions and 100% of the cases for overflow. This finding demonstrates the potential of the proposed method to improve retrospective streamflow estimates in data-scarce scenarios by leveraging observed data from downstream gauges, along with comprehensive physical and meteorological information of the watershed.

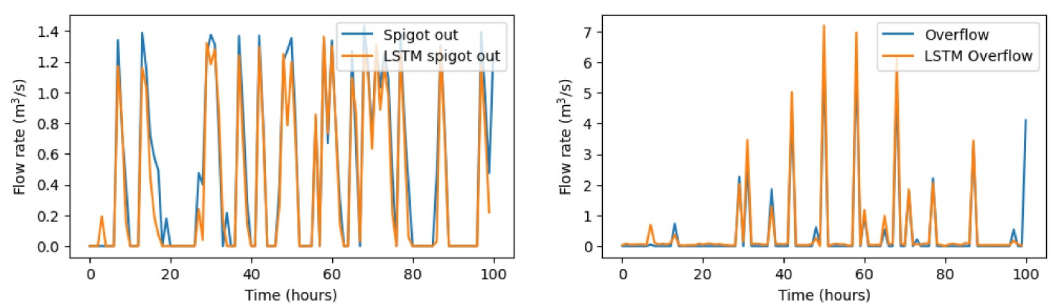
4.3. Analysis of Extreme Flow Conditions

A follow-up experiment was conducted to assess the model's performance when exposed to both low-flow and high-flow extremes. This investigation addresses whether the combined bucket network approach remains effective beyond moderate or "normal" flows, providing additional insight into the model's robustness under more challenging hydrologic regimes.

A new training data set was created with 2,000 synthetic bucket networks–each composed of two buckets with distinct physical and meteorological characteristics–containing 10,000 hourly records per bucket (approximately 417 days of simulation), to capture a wider range of flow behaviors. Precipitation parameters were also broadened



(a) Bucket 1



(b) Bucket 2

Figure 3. Spigot flow and overflow predictions for one of the networks in the validation data set.

Table 2
Comparison of Average Performance Metrics for Both Approaches on the Test Data Set (i.e., Basins/Networks Not Used for Training)

	Spigot flow			Overflow		
Approach	RMSE	MAE	NSE	RMSE	MAE	NSE
Individual bucket	0.194	0.149	0.788	0.396	0.149	0.876
Combined bucket network	**0.144**	**0.109**	**0.886**	**0.210**	**0.087**	**0.961**
% Improvement	26.06%	26.72%	12.44%	46.87%	41.88%	9.75%

to incorporate extremes in flow generation. Using the same hyperparameters identified in the previous optimization (see Appendix B, Section B5), an LSTM was trained on this expanded data set.

To evaluate performance on flow extremes, three additional test sets were generated which were not included in the training. Each comprised 300 synthetic bucket networks (two buckets each) with 1,500 hourly records (about 62.5 days of simulation). The first set, referred to as "Normal Flows," had a distribution of bucket networks similar to that of the training set and served as a benchmark. The second set, called the "Low-Flow Scenario," contained mostly low precipitation rates and consistently small flow magnitudes. Finally, the "High-Flow Scenario" involved higher precipitation intensities and volumes, leading to markedly larger spigot flows and more frequent overflow events. As shown in Table 3, the model's NSE was used for comparison across these three data sets. Note that RMSE and MAE were not employed here, as their absolute error measures are less comparable when the flow magnitudes differ substantially among scenarios.

As seen in Table 3, the model achieves moderate to high NSE values across all conditions, though performance varies by flow regime. Notably, spigot flow predictions weaken in low-flow and high-flow scenarios, indicating that—even with a broader training range—extremes are more challenging to capture. Meanwhile, overflow predictions are less accurate when these events are rare (low-flow scenario) but remain strong under consistently higher flows. These outcomes emphsize the value of including representative examples for each flow regime during training and suggest that targeted strategies, such as data augmentation or specialized loss functions, could further enhance predictions in extreme cases.

Overall, the results indicate that the combined bucket network approach preserves a reasonable level of accuracy in diverse flow conditions, but that its predictions are strongest when streamflow falls within a more moderate range. Nonetheless, even under low-flow and high-flow extremes, the combined approach consistently outperformed the individual bucket model, as shown in Table 3. Further experimentation—potentially including specialized training strategies for rare events or integrating more advanced hydrologic routing—may help to enhance predictive skill for extreme flows. This additional analysis reinforces the model's promise, while also highlighting areas for refinement as it is adapted to more complex or real-world basins.

Although the results obtained are promising, it is recognized that this study represents an initial proof of concept. Therefore, future research should focus on expanding the data sets to include a greater variety and volume of synthetic data. Exploration of this approach in larger and more diverse hydrologic networks will allow the model to be validated and fine-tuned, although this will involve higher computational requirements. In addition, once a sufficiently robust model has been created using synthetic data, the next step should be the integration of real data

Table 3
Nash–Sutcliffe Efficiency Values for Spigot Flow and Overflow in Three Test Data Sets (Normal, Low-Flow, and High-Flow) Using the Newly Trained Long Short-Term Memory Model on an Expanded Data Set

	Spigot flow NSE			Overflow NSE		
	Flow Scenario			Flow Scenario		
Approach	Normal	Low	High	Normal	Low	High
Individual bucket	0.739	0.683	0.655	0.867	0.452	0.823
Combined bucket network	**0.834**	**0.793**	**0.755**	**0.956**	**0.755**	**0.935**
% Improvement	12.92%	16.04%	15.32%	10.33%	66.92%	13.68%

from fully monitored catchments to advance the practical application of the model in the creation of Synthetic Gauges in real-world contexts.

5. Conclusions

This study introduces a novel methodology using LSTM to improve historical flow estimates in unmonitored portions of a synthetic watershed. The proposed model demonstrated strong predictive performance for both spigot flow and overflow, supporting the suitability of LSTM-based models for interpreting complex hydrologic time series. These results emphasize the potential of deep learning approaches, particularly LSTMs, in synthetic experiments designed to emulate the challenges of data-scarce environments. Future research should extend this work to real-world gauge data to further validate the applicability of this method.

The integration of the Leaky Bucket model into LSTM networks presents a promising avenue for future hydroinformatics research, especially in regions with limited monitoring infrastructure. Future research could explore more sophisticated flow routing methods to enhance the model's predictive capability for various hydrologic phenomena, further improving its application in ungauged basins.

This study demonstrates that incorporating combined downstream flow data, along with the physical and meteorological characteristics of all synthetic watersheds, significantly improves the model's ability to retrospectively estimate streamflow in unmonitored portions of a synthetic watershed, compared to the traditional approach, which relies solely on individual watershed information for such estimates. This process is conceptually analogous to signal separation techniques used in fields like audio processing, where combined signals are disentangled to isolate individual components. Similarly, our method leverages downstream flow data to better estimate the "individual signals" of upstream flow components within the watershed.

Furthermore, a novel conceptualization of watershed networks is proposed using two buckets. This representation allows the generalization of the LSTM-based approach to fill in surrogate gauges in a retrospective record of a network of interconnected gauges. A proof of concept of such an approach is presented using synthetic hydrographs from bucket models, demonstrating that any unmonitored portion of a watershed can be conceptually modeled as a pair: one bucket representing the monitored downstream segment and another bucket representing any of its unmonitored tributaries. While this simplified bucket model is not intended to directly simulate real-world watersheds, it provides a useful framework for demonstrating the utility of the combined training process over an individual watershed approach. This network configuration allows the model to simulate not only water flows in individual segments but also their interaction throughout the entire watershed. This paired representation offers a more practical and generalizable way to capture key hydrologic behavior on a broader scale by focusing on the essential connection between upstream and downstream segments, rather than attempting to fully model the complexity inherent in an entire watershed network.

Additionally, an analysis of low-flow and high-flow extremes showed that while the model's accuracy diminishes somewhat under these less frequent flow conditions, it still remains effective across a wide hydrologic spectrum. This outcome reinforces the value of ensuring sufficient representation of extreme events in training data and suggests that approaches like data augmentation or specialized loss functions could further improve performance at the tails of the flow distribution.

This approach not only illustrates the potential for enhanced flow estimates in data-limited scenarios but also provides a generalizable methodology for hydrological analysis in watershed networks, establishing a foundation for applications where downstream observations are available. By intentionally employing synthetic data within a controlled environment, this study demonstrates the feasibility of the framework and suggests opportunities for broader implementation in real-world networks.

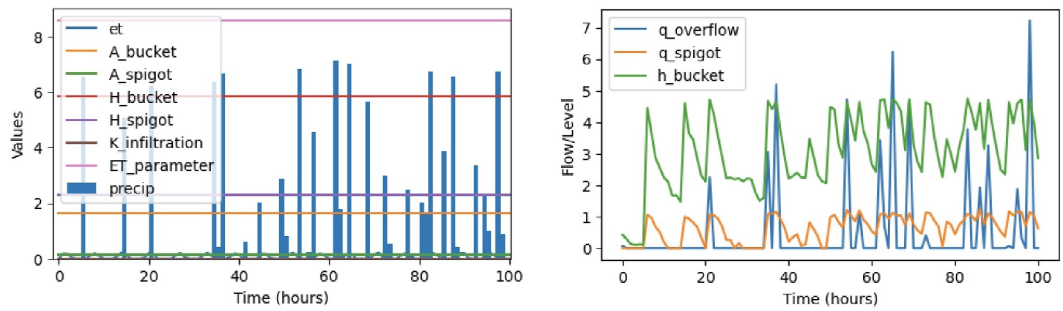
Future research should aim to expand the data sets, apply the methodology to larger and more diverse hydrologic networks, and explore the inclusion of additional parameters that may influence flow rates. Extending the model with real data from fully monitored catchments would further support the creation of Synthetic Gauges for practical applications.

By advancing LSTM applications in hydrology, this study contributes to the broader field of water resources management, helping to mitigate natural phenomena and protect ecosystems.

Appendix A: Synthetic Data Set Generation for the Bucket Networks

A1.1. Bucket Model Outputs

To understand the individual dynamics within the bucket network, Figure A1 shows the results produced by the Leaky Bucket model for a single bucket. The first graph (Figure A1a) displays the simulated meteorological inputs to the model, such as precipitation (*precip*) and evapotranspiration (*et*), as well as the physical characteristics of the bucket: bucket area (*A_bucket*), spigot area (*A_spigot*), bucket height (*H_bucket*), spigot height (*H_spigot*), infiltration parameter (*K_infiltration*), and evapotranspiration parameter (*ET_parameter*). The second graph (Figure A1b) shows the outputs produced by the model, including bucket flow (*q_spigot*), overflow (*q_overflow*), and water level inside the bucket (*h_bucket*). These output time series represent synthetic hydrographs for spigot flow and overflow, which are analyzed throughout this study.



(a) Model inputs (b) Model outputs

Figure A1. Inputs and outputs simulated by the Leaky Bucket model for a single bucket. (a) Simulated meteorological inputs and physical characteristics of the bucket model. Units for *precip* and *et* are in *mm*, *A_bucket* and *A_spigot* are in m^2, and *H_bucket* and *h_spigot* are in *m*, while *K_infiltration* and *ET_parameter* are dimensionless. (b) Simulated outputs. Units for *q_spigot* and *q_overflow* are in m^3/s, while *h_bucket* is in *m*.

A1.2. Combined Network Flow

Figure A2 illustrates the resulting combined streamflow in a network of buckets. This visualization shows how the individual flows from each bucket are integrated through the null routing function to form the total downstream flow of the network (*q_total_output*). The total flow (*q_total*) is computed as the sum of the flows through the spigot (*q_spigot*) and the overflow (*q_overflow*) for each bucket.

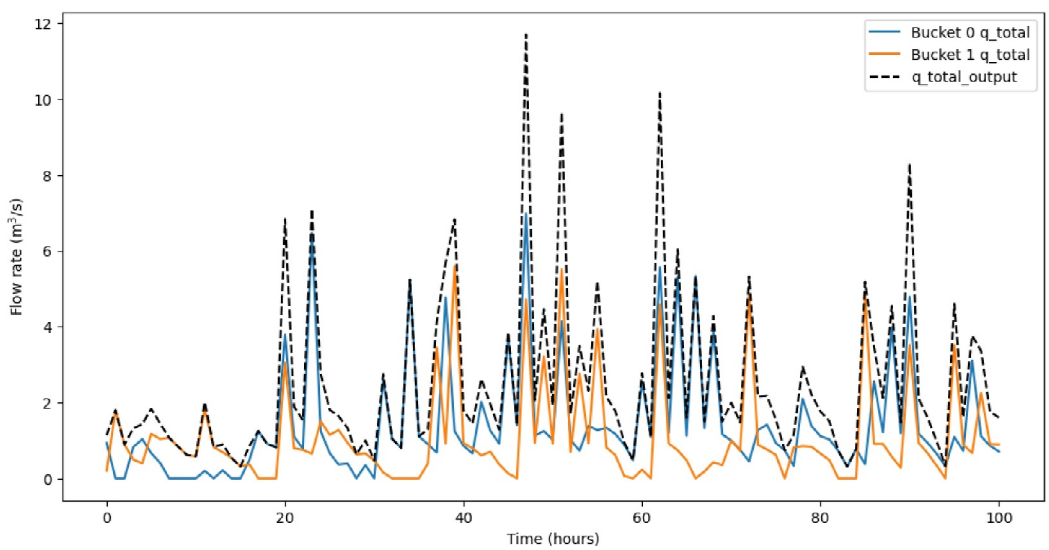


Figure A2. Combined streamflows in a network of buckets. The graph shows the individual total flows (*q_total*) for Bucket 0 and Bucket 1, as well as the overall total downstream flow (*q_total_output*).

Three data sets were created for this study: a training data set, a validation data set, and a test data set. The training data set consists of 350 unique bucket networks, each composed of two buckets with independently sampled physical parameters and meteorological forcings. Each bucket has 7,000 records, simulating hourly measurements over 291.7 days. The validation data set contains 204 networks, also with two buckets each, and each bucket has 1,500 records, simulating a period of 62.5 days. Additionally, a test data set of the same size and structure as the validation data set was created to further measure the performance of the optimized model. These synthetic data sets, while simplified, enable controlled experimentation to assess the feasibility of the combined downstream flow approach in ungauged segments.

Appendix B: LSTM Model Implementation

B1.1. LSTM Cell

The architecture of the LSTM cell can be seen in Figure B1. The gates are specialized mechanisms that effectively control the flow of information, allowing the LSTM cell to retain or discard information over time. A typical LSTM cell consists of three main gates: the forget gate, the input gate, and the output gate. In addition, the cell has two states that it transmits over time: the cell state (C_t) and the hidden state (h_t) (Hochreiter & Schmidhuber, 1997).

The forget gate decides which information should be discarded from the cell state. This decision is based on the current input and the previous hidden state. Mathematically, it is expressed as:

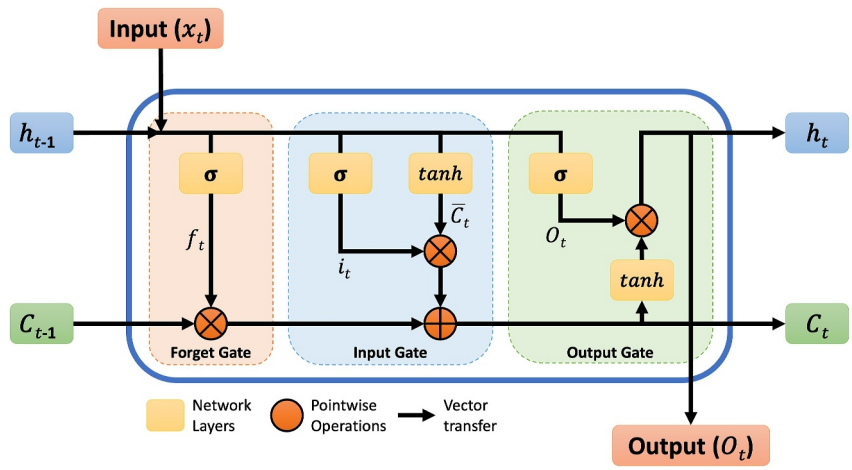


Figure B1. The architecture of the long short-term memory cell. Adapted from (Xu et al., 2020).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{B1}$$

where f_t is the output of the forget gate, *sigma* denotes the sigmoid function, W_f is the weights matrix for the forget gate, h_{t-1} is the previous hidden state, x_t is the current input, and b_f is the bias for the forget gate (Le et al., 2019).

The input gate controls the flow of new information to the cell state. It consists of two parts: a sigmoid layer that decides which values are to be updated and an activation function tanh that creates a vector of new candidate values, which could be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{B2}$$

$$\tilde{C}t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{B3}$$

here, i_t is the output of the input gate, which regulates how much new information is allowed into the cell state, while $\tilde{C}_t$ represents the candidate values for the cell state that may potentially update it. W_i and W_C are the weights

matrices for the input gate and the candidate state, respectively, and b_i and b_C are the corresponding biases (Le et al., 2019).

The cell state is updated by multiplying the previous cell state by f_t, thus removing the information that the forgetting gate has decided to discard. Then, the product of i_t and $\tilde{C}_t$ is summed to add new information.

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{B4}$$

where C_t is the updated state of the cell (Le et al., 2019).

Finally, the output gate decides which part of the cell state is to be transmitted to the hidden state, which is the output of the LSTM cell.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{B5}$$

$$h_t = o_t \times \tanh(C_t) \tag{B6}$$

where o_t is the output of the output gate, W_o is the weights matrix for the output gate, b_o is the bias of the output gate and h_t is the updated hidden state (Le et al., 2019).

These gates and operations allow LSTMs to regulate the flow of information over time, making them particularly suitable for learning long-term dependencies in data sequences, avoiding problems such as vanishing or exploding gradient.

B1.2. LSTM Network Architecture

The LSTM network designed in this study consists of three types of layers, specifically adapted to address the complexity of hydrological time series. As seen in Figure B2, the architecture is composed of:

1. *Input Layer*: Responsible for receiving the sequential input data, including variables such as precipitation, evapotranspiration, and combined downstream flow, along with the characteristics of the buckets.

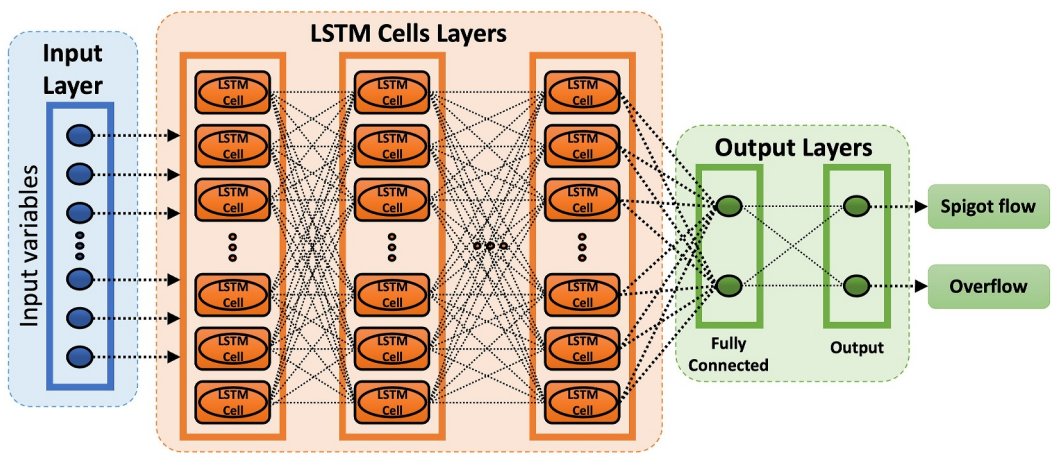


Figure B2. Architecture of the long short-term memory (LSTM) Network. The input variables correspond to the features simulated by each Leaky Bucket Model within the bucket network, while the output variables are the predictions of spigot flow and overflow that the LSTM model makes for each bucket.

2. *LSTM Cells Layers*: It was determined through hyperparameter optimization that this layer should be composed of 36 cells and 60 hidden layers. These cells are responsible for the network's ability to remember and learn from long-term temporal information.

It should be noted that the PyTorch 2.0.1 library was used to implement the LSTM network. In the standard PyTorch LSTM models, the transformations between the input layer and the LSTM cells, such as embedding the input data and applying the necessary weight matrices to compute the cell states and gates, are handled internally by the library. These transformations prepare the input data for processing by the LSTM cells without the need for an additional fully connected layer (PyTorch, 2023).

3. *Output Layers*: Including a fully connected layer and the output layer. The fully connected layer transforms the information processed by the LSTM cells into a format suitable for the final prediction of the spigot flow and overflow of each bucket.

The model considers a sequence length of 24, which means that it observes and learns from the last 24 observations before making a prediction. In the simulation done by the Leaky Bucket Model, each observation has a frequency of one hour, so a sequence length of 24 captures one full day of hydrological behavior. This sequence length was chosen to align with the daily hydrological cycle simulated in the data. Preliminary tests with shorter and longer sequences confirmed that 24 hr provided a consistent balance between predictive performance and temporal interpretability.

For model training, the Adam optimization algorithm is used with a learning rate varying linearly from 0.1 to 0.001 over 30 epochs (epoch number was determined through hyperparameter optimization). This technique helps to adjust the network weights efficiently, improving the convergence of the model (Goodfellow et al., 2016).

Additionally, during training, the model uses the Mean Squared Error (MSE) as a loss function to guide the optimization process by calculating the discrepancy between predicted and actual values. As shown in Equation B7, the MSE measures the average of the squares of the differences between the predicted and actual values, which provides a quantitative measure of the model's accuracy. The MSE is sensitive to variations in the data, penalizing large errors more severely than small errors. This feature makes it particularly useful in regression tasks, as is the case in this study, which seeks to predict dependent variables (spigot flow and overflow), given independent variables (simulated bucket characteristics). The use of MSE as a loss function seeks to focus the model on minimizing significant differences, in order to achieve accurate and reliable predictions during the learning process (Goodfellow et al., 2016).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{B7}$$

B1.3. Data Preprocessing and Postprocessing

Before feeding the data to the LSTM network, the input values are normalized to ensure that they are at a scale suitable for being processed by the network. In the implemented model, the *StandardScaler* from the *scikit-learn* library is used to normalize the input data (scikit-learn developers, 2023). This method applies Equation B8 by adjusting the data to have a mean of zero and a standard deviation of one. Normalization helps to improve training efficiency and avoid problems such as vanishing or exploding gradients (Goodfellow et al., 2016).

$$x_{\mathrm{norm}} = \frac{x - \mu}{\sigma} \tag{B8}$$

where x is the original value μ is the mean of training set, and σ is the standard deviation of the training set.

After processing by the LSTM network and obtaining the predictions, a denormalization step is performed to transform the network outputs back to their original scale Equation B9. This step is necessary for results interpretation and comparison. The denormalization is performed using the normalization parameters (mean and standard deviation) initially applied, that is, those of the training set before being processed by the LSTM network.

$$x_{\mathrm{original}} = x_{\mathrm{norm}} \cdot \sigma + \mu \tag{B9}$$

B1.4. Pseudo-Code for LSTM Network

The following algorithm shows the pseudo-code of the training process of the LSTM network:

```
Input:
  Input data matrix: [T, m] //T: Time series - m: Input variables (buckets features)
  Target data matrix: [T, g] //g: Output variables (Spigot flow and Overflow)
Architecture Definition:
  Number of LSTM layers: 60
  Number of LSTM cells per layer: 36
Initial Parameters:
  LSTM parameters (weights, biases)
  Initialize cell state and hidden state to zero for each cell
LSTM Cells:
  For each timestep t in T:
    For each layer l in 60 LSTM layers:
      For each cell k in 36 LSTM cells:
        Update cell state and hidden state using LSTM equations
        Generate cell output
  Compile outputs of LSTM cells: [T, 36]
Output Layer:
  Fully Connected Layer (fc_1):
    Transform [T, 36] to [T, 2] //2 output features: Spigot flow and Overflow
Loss Function:
  Compute MSE between predicted output and actual target data
Weights Updating:
  Update network weights using Adam optimization algorithm based on the loss value
```

Where T is the batch size of the training data; m is the number of input variables; and g is the number of output variables.

B1.5. Hyperparameter Optimization of the LSTM Model

The model hyperparameter optimization process focused on testing different configurations of the LSTM network size. Specifically, by varying the number of hidden layers, the depth of the model was adjusted. At the same time, its complexity was fine-tuned by varying the number of LSTM cells in each layer. Likewise, the learning time of the model was set by varying the number of training epochs. Table B1 summarizes 18 different configurations

Table B1
Hyperparameter Optimization Results

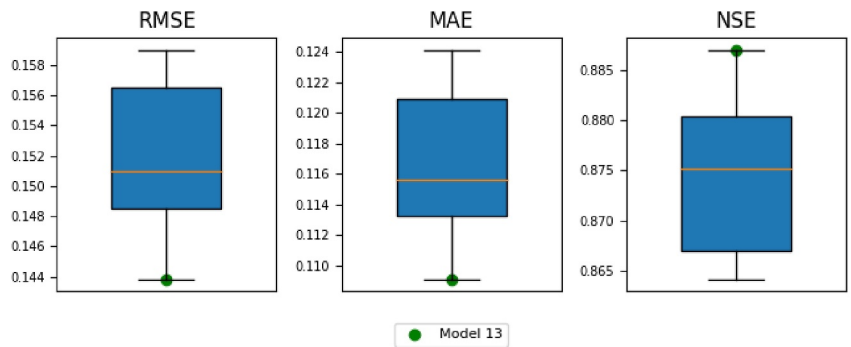
	LSTM			Spigot flow			Overflow		
Model	Layers	Cells	Epochs	RMSE	MAE	NSE	RMSE	MAE	NSE
1	8	16	20	0.151	0.114	0.880	0.269	0.113	0.939
2	16	80	80	0.156	0.122	0.865	0.233	0.094	0.952
3	24	32	20	0.151	0.115	0.875	0.236	0.124	0.953
4	24	64	20	0.157	0.121	0.864	0.243	0.108	0.950
5	32	24	20	0.159	0.124	0.865	0.244	0.113	0.948
6	32	32	20	0.151	0.115	0.875	0.236	0.124	0.953
7	32	48	20	0.145	0.110	0.887	0.237	0.110	0.952
8	32	160	20	0.145	0.110	0.884	0.226	0.107	0.957
9	48	24	20	0.159	0.124	0.865	0.244	0.113	0.948
10	48	24	30	0.151	0.116	0.880	0.247	0.116	0.947

Table B1
Continued

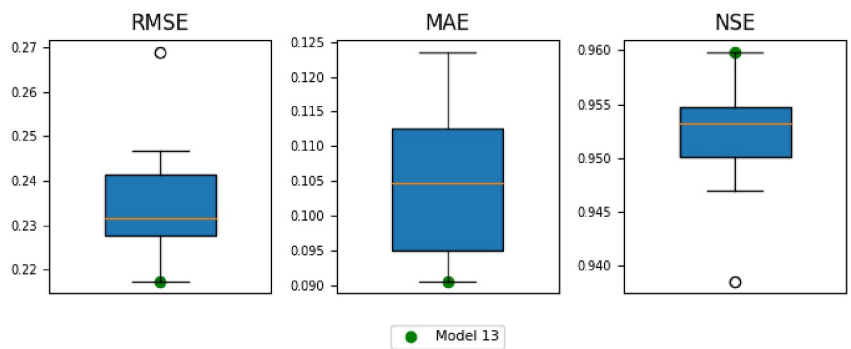
Model	Layers	LSTM Cells	Epochs	Spigot flow RMSE	Spigot flow MAE	Spigot flow NSE	Overflow RMSE	Overflow MAE	Overflow NSE
11	60	30	30	0.157	0.122	0.867	0.230	0.092	0.955
12	60	36	20	0.148	0.113	0.880	0.230	0.103	0.954
13	**60**	**36**	**30**	**0.144**	**0.109**	**0.887**	**0.217**	**0.091**	**0.960**
14	60	36	40	0.155	0.120	0.867	0.224	0.091	0.955
15	80	36	20	0.148	0.113	0.880	0.230	0.103	0.954
16	150	75	36	0.147	0.112	0.882	0.228	0.098	0.955
17	150	75	40	0.153	0.117	0.873	0.228	0.096	0.955
18	150	100	36	0.156	0.120	0.870	0.224	0.095	0.955

tested and the average of the metrics obtained for the flow through the spigot and overflow predictions of every bucket in the 204 networks in the validation data set.

Model 13, with 60 hidden layers of 36 LSTM cells each, trained for 30 epochs, proved to be the best at predicting spigot flow and overflow. This superior performance is statistically evident, as Model 13 is more than 1.5 standard deviations above the average performance metrics of other models for spigot flow prediction, and more than 1.3 standard deviations above for overflow prediction. Figure B3 illustrates how Model 13 excels in RMSE, MAE, and NSE for both spigot flow and overflow predictions.



(a) Spigot flow prediction metrics

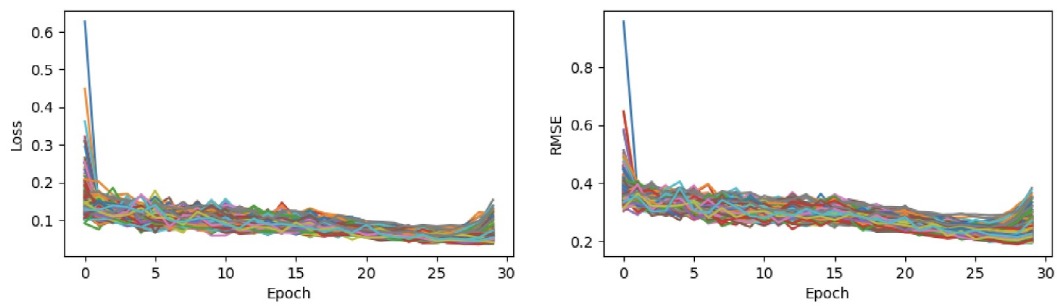


(b) Overflow prediction metrics

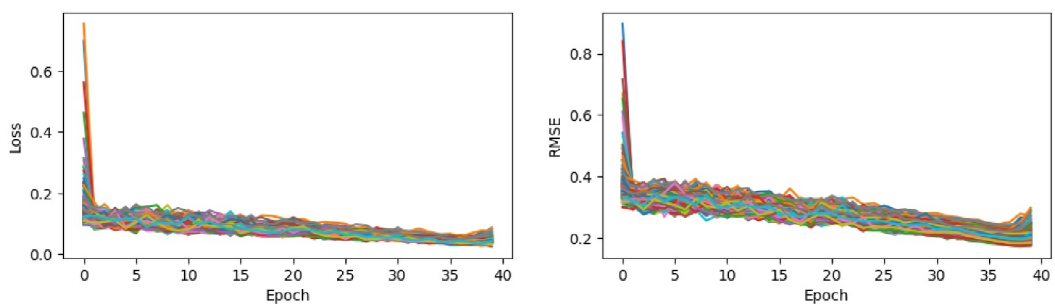
Figure B3. Comparison of performance metrics among the tested models. Model 13 obtains the best results in statistical terms for both spigot flow and overflow predictions. (a) Shows the performance metrics for spigot flow prediction. (b) Shows the performance metrics for overflow prediction. The poor performance of Model 1 in *RMSE* and *NSE* for overflow prediction makes it an outlier.

To further support this comparison, a *t*-test was conducted treating model 13 as a group of one and the remaining 17 models as a second group. The test showed that the performance of model 13 is significantly better across all six metrics (RMSE, MAE, and NSE for both spigot and overflow) with p-values <0.0001. Additionally, individual *t*-tests were run comparing model 13 to each of the other configurations. In these tests, the metrics for each model were calculated across all individual buckets in the validation set, enabling estimation of within-model variance. At a significance level of $\alpha = 0.05$, model 13 outperformed models 3, 4, 5, 6, and 9 across all six metrics; models 1, 2, 10, 11, 12, 15, and 17 in five metrics; models 14, 16, and 18 in four metrics; model 7 in three metrics; and model 8 in one metric.

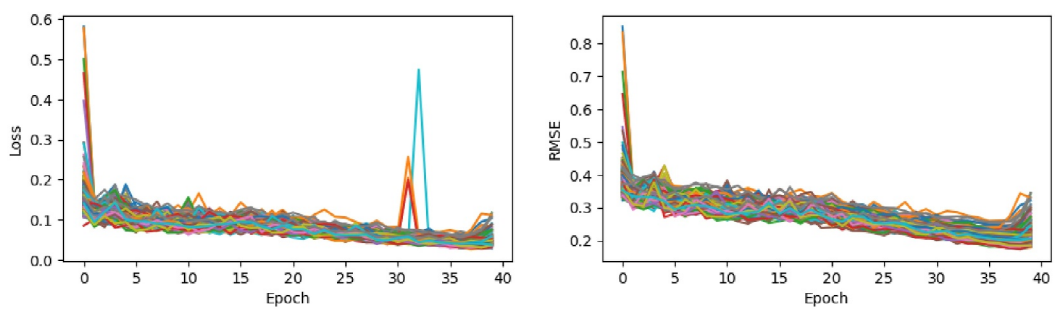
Consequently, as shown in Figure B4a, the learning curves of model 13 indicate that convergence occurred approximately after epoch 28. Compared to Model 16, a more complex and deeper model of 150 hidden layers and 75 LSTM cells (Figure B4b), it is observed that convergence is reached in the first 2 or 3 epochs, after which model 16 does not appear to learn anything new from the data. Similarly, compared to model 14, an equally complex and deep model, but with 33% more training time, that is, for 40 epochs, it is observed that the convergence of the model begins to lose after epoch 30 (Figure B4c).



(a) Model 13. With 60 Layers, 36 LSTM Cells and 30 Epochs



(b) Model 16. With 150 Layers, 75 LSTM Cells and 40 Epochs



(c) Model 14. With 60 Layers, 36 LSTM Cells and 40 Epochs

Figure B4. Learning curves during training of three different models. The plots show the evolution of the loss function and the root mean square error per epoch for each record of the training data set.

The above corroborates the fact that in deep learning models, better results are obtained when there is a balance in the complexity of the model. Not necessarily the most complex model is the best; in this case, the deepest models with a moderate number of LSTM cells per hidden layer demonstrated the ability to extract more information from the training data, which is a common behavior in deep learning techniques (Goodfellow et al., 2016).

Data Availability Statement

The code used to develop the Leaky Bucket model and the basin networks to generate the synthetic data utilized in this study, as well as the LSTM model employed to separate the combined downstream flow of the major rivers into the individual upstream flows of their tributaries, is publicly available at https://doi.org/10.5281/zenodo.14028825 (Ramírez Molina et al., 2024).

Acknowledgments
Funding for this project was provided in part by the NOAA, awarded to the Cooperative Institute for Research to Operations in Hydrology (CIROH) through the Cooperative Agreement with The University of Alabama (NA22NWS4320003). The Cooperative Institute for Research to Operations in Hydrology operates under Cooperative Agreement #NA22NWS4320003 from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce. The statements, findings, conclusions, and recommendations presented are those of the authors and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration or the Department of Commerce.

References

Adams, S. K. (2021). *Improving hydrologic modeling of ungaged basins to support environmental flow management in a heterogeneous region*. Colorado State University. Retrieved from https://core.ac.uk/download/524898115.pdf

Cosgrove, B., Gochis, D., Flowers, T., Dugger, A., Ogden, F., Graziano, T., et al. (2024). NOAA's national water model: Advancing operational hydrology through continental-scale modeling. *JAWRA Journal of the American Water Resources Association*, *60*(2), 247–272. https://doi.org/10.1111/1752-1688.13184

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., et al. (2018). Looking to listen at the cocktail party. *ACM Transactions on Graphics*, *37*(4), 1–11. https://doi.org/10.1145/2F3197517.3201357

Fan, H., Jiang, M., Xu, L., Zhu, H., Cheng, J., & Jiang, J. (2020). Comparison of long short term memory networks and the hydrological model in runoff simulation. *Water*, *12*(1), 175. https://doi.org/10.3390/w12010175

Feng, D., Beck, H., Lawson, K., & Shen, C. (2023). The suitability of differentiable, physics-informed machine learning hydrologic models for ungauged regions and climate change impact assessment. *Hydrology and Earth System Sciences*, *27*(12), 2357–2373. https://doi.org/10.5194/hess-27-2357-2023

Fisher, C. K., Pan, M., & Wood, E. F. (2020). Spatiotemporal assimilation–interpolation of discharge records through inverse streamflow routing. *Hydrology and Earth System Sciences*, *24*(1), 293–305. https://doi.org/10.5194/hess-2018-109

Frame, J. M., Lee, Q., maoyab, & Hernandez Rodriguez, L. (2024). NWC-CUAHSI-Summer-Institute/deep_bucket_lab: Zenodo re-try (version 1.2) [Computer software]. *Zenodo*. https://doi.org/10.5281/zenodo.14538196

Gerhart, A. L., Hochstein, J. I., & Gerhart, P. M. (2021). *Munson, young and Okiishi's fundamentals of fluid mechanics* (9th ed.). Wiley and Sons.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from http://www.deeplearningbook.org

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

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., et al. (2013). A decade of predictions in ungauged basins (pub)—A review. *Hydrological Sciences Journal*, *58*(6), 1198–1255. https://doi.org/10.1080/02626667.2013.803183

Hu, Y., Yan, L., Hang, T., & Feng, J. (2020). *Stream-flow forecasting of small rivers based on lstm*. ArXiv, abs/2001.05681. Retrieved from https://api.semanticscholar.org/CorpusID:210701503

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Jiang, S., Zheng, Y., & Solomatine, D. (2020). Improving AI system awareness of geoscience knowledge: Symbiotic integration of physical approaches and deep learning. *Geophysical Research Letters*, *47*(13), e2020GL088229. https://doi.org/10.1029/2020GL088229

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., & Nearing, G. S. (2019). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, *55*(12), 11344–11354. https://doi.org/10.1029/2019WR026065

Krauße, T. (2007). Development of a class framework for flood forecasting. Techn. Univ., Fakultät Informatik.

Le, X.-H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, *11*(7), 1387. https://doi.org/10.3390/w11071387

MacBean, N., Scott, R. L., Biederman, J. A., Ottlé, C., Vuichard, N., Ducharne, A., et al. (2020). Testing water fluxes and storage from two hydrology configurations within the orchidee land surface model across us semi-arid sites. *Hydrology and Earth System Sciences*, *24*(11), 5203–5230. https://doi.org/10.5194/hess-24-5203-2020

Nash, J. (1957). *The form of the instantaneous unit hydrograph* (Vol. 45, pp. 114–121). Publications—International Association of Hydrological Sciences.

Nash, J., & Sutcliffe, J. (1970). River flow forecasting through conceptual models part I—A discussion of principles. *Journal of Hydrology*, *10*(3), 282–290. https://doi.org/10.1016/0022-1694(70)90255-6

National Oceanic and Atmospheric Administration (NOAA). (2016). The national water model. Retrieved from https://water.noaa.gov/about/nwm

National Oceanic and Atmospheric Administration (NOAA). (2023). Leaky bucket model. Retrieved from https://www.cpc.ncep.noaa.gov/soilmst/global/

Perera, U., Coralage, D., Ekanayake, I., Alawatugoda, J., & Meddage, D. (2024). A new Frontier in streamflow modeling in ungauged basins with sparse data: A modified generative adversarial network with explainable AI. *Results in Engineering*, *21*, 101920. https://doi.org/10.1016/j.rineng.2024.101920

PyTorch. (2023). Lstm — Pytorch 2.0 documentation. Retrieved from https://pytorch.org/docs/2.0/generated/torch.nn.LSTM.html#torch.nn.LSTM

Ramírez Molina, A. A., Bezak, N., Tootle, G., Wang, C., & Gong, J. (2023). Machine-learning-based precipitation reconstructions: A study on Slovenia's Sava River Basin. *Hydrology*, *10*(11), 207. https://doi.org/10.3390/hydrology10110207

Ramírez Molina, A. A., Frame, J. M., Halgren, J., & Gong, J. (2024). *Synthetic stream gauges: Code for "A proof of concept for improving estimates of Ungauged Basin streamflow via an LSTM-based synthetic network simulation approach"*. Zenodo. https://doi.org/10.5281/zenodo.14028825

scikit-learn developers. (2023). sklearn.preprocessing.Standardscaler–Scikit-learn. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

Sivapalan, M., Takeuchi, K., Franks, S. W., Gupta, V. K., Karambiri, H., Lakshmi, V., et al. (2003). Iahs decade on predictions in ungauged basins (pub), 2003–2012: Shaping an exciting future for the hydrological sciences. *Hydrological Sciences Journal*, *48*(6), 857–880. https://doi.org/10.1623/hysj.48.6.857.51421

Terink, W., Lutz, A. F., Simons, G. W. H., Immerzeel, W. W., & Droogers, P. (2015). Sphy v2.0: Spatial processes in hydrology. *Geoscientific Model Development*, *8*(7), 2009–2034. https://doi.org/10.5194/gmd-8-2009-2015

U.S. Geological Survey. (2024). National hydrography dataset. (Accessed: Insert access date here). Retrieved from https://www.usgs.gov/national-hydrography/national-hydrography-dataset

Vrugt, J. A., Gupta, H. V., Dekker, S. C., Sorooshian, S., Wagener, T., & Bouten, W. (2006). Application of stochastic parameter optimization to the Sacramento Soil Moisture Accounting model. *Journal of Hydrology*, *325*(1), 288–307. https://doi.org/10.1016/j.jhydrol.2005.10.041

Wang, Y.-H., & Gupta, H. V. (2024). A mass-conserving-perceptron for machine-learning-based modeling of geoscientific systems. *Water Resources Research*, *60*(4), e2023WR036461. https://doi.org/10.1029/2023WR036461

Wilbrand, K., Taormina, R., ten Veldhuis, M.-C., Visser, M., Hrachowitz, M., Nuttall, J., & Dahm, R. (2023). Predicting streamflow with LSTM networks using global datasets. *Frontiers in Water*, *5*. https://doi.org/10.3389/frwa.2023.1166124

Xu, W., Jiang, Y., Zhang, X., Li, Y., Zhang, R., & Fu, G. (2020). Using long short-term memory networks for river flow prediction. *Hydrology Research*, *51*(6), 1358–1376. https://doi.org/10.2166/nh.2020.026