Uncertainty Quantification of Machine Learning Models to Improve Streamflow Prediction Under Changing Climate and Environmental Conditions

Dan Lu[1], Siyan Liu[2], Scott L Painter[3], Natalie A Griffiths[1], and Eric M Pierce[1]

[1]Oak Ridge National Laboratory (DOE)
[2]Oak Ridge National Laboratory
[3]Oak Ridge National Laboratory, USA

November 22, 2022

Abstract

Machine learning (ML) models, and Long Short-Term Memory (LSTM) networks in particular, have demonstrated remarkable performance in streamflow prediction and are increasingly being used by the hydrological research community. However, most of these applications do not include uncertainty quantification (UQ). ML models are data driven and may suffer from large extrapolation errors when applied to changing climate/environmental conditions. UQ is required to ensure model trustworthiness, improve understanding of data limits and model deficiencies, and avoid overconfident predictions in extrapolation. Here, we propose a novel UQ method, called PI3NN, to quantify prediction uncertainty of ML models and integrate the method with LSTM networks for streamflow prediction. PI3NN calculates Prediction Intervals by training 3 Neural Networks and uses root-finding methods to determine the interval precisely. Additionally, PI3NN can identify out-of-distribution (OOD) data in a nonstationary condition to avoid overconfident prediction. We apply the proposed PI3NN-LSTM method in both the snow-dominant East River Watershed in the western US and the rain-driven Walker Branch Watershed in the southeastern US. Results indicate that for the prediction data (which have similar features as the training data), PI3NN precisely quantifies the prediction uncertainty with the desired confidence level; and for the OOD data where the LSTM network fails to make accurate predictions, PI3NN produces a reasonably large uncertainty bound indicating the untrustworthy result to avoid overconfidence. PI3NN is computationally efficient, reliable in training, and generalizable to various network structures and data with no distributional assumptions. It can be broadly applied in ML-based hydrological simulations for credible prediction.

Uncertainty Quantification of Machine Learning Models to Improve Streamflow Prediction Under Changing Climate and Environmental Conditions

Siyan Liu[1], Dan Lu[1], Scott L. Painter[2], Natalie A. Griffiths[2], Eric M. Pierce[2]

[1]Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
[2]Environmental Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA

Key Points:

- We developed an uncertainty quantification method to quantify machine learning model prediction uncertainty
- We integrated the method with Long Short-Term Memory networks for streamflow predictions in both snow-dominant and rain-driven watersheds
- The method precisely quantifies the prediction uncertainty and avoids overconfident projections in new climate conditions

Corresponding author: Dan Lu, `lud1@ornl.gov`

Abstract

Machine learning (ML) models, and Long Short-Term Memory (LSTM) networks in particular, have demonstrated remarkable performance in streamflow prediction and are increasingly being used by the hydrological research community. However, most of these applications do not include uncertainty quantification (UQ). ML models are data driven and may suffer from large extrapolation errors when applied to changing climate/environmental conditions. UQ is required to ensure model trustworthiness, improve understanding of data limits and model deficiencies, and avoid overconfident predictions in extrapolation. Here, we propose a novel UQ method, called PI3NN, to quantify prediction uncertainty of ML models and integrate the method with LSTM networks for streamflow prediction. PI3NN calculates Prediction Intervals by training 3 Neural Networks and uses root-finding methods to determine the interval precisely. Additionally, PI3NN can identify out-of-distribution (OOD) data in a nonstationary condition to avoid overconfident prediction. We apply the proposed PI3NN-LSTM method in both the snow-dominant East River Watershed in the western US and the rain-driven Walker Branch Watershed in the southeastern US. Results indicate that for the prediction data (which have similar features as the training data), PI3NN precisely quantifies the prediction uncertainty with the desired confidence level; and for the OOD data where the LSTM network fails to make accurate predictions, PI3NN produces a reasonably large uncertainty bound indicating the untrustworthy result to avoid overconfidence. PI3NN is computationally efficient, reliable in training, and generalizable to various network structures and data with no distributional assumptions. It can be broadly applied in ML-based hydrological simulations for credible prediction.

1 Introduction

Accurate prediction of streamflow is critical for short-term flood risk mitigation and long-term water resources management necessary to advance agricultural and economic development. Machine learning (ML) models demonstrate excellent performance in streamflow prediction and are being used more often as a tool by the hydrological community (Rasouli et al., 2012; Shortridge et al., 2016; Tongal & Booij, 2018; Kratzert et al., 2018, 2019; Feng et al., 2020; Shamshirband et al., 2020; Konapala et al., 2020; Xu & Liang, 2021; Lu et al., 2021). However, most of these applications do not include uncertainty quantification (UQ) and generally only produce deterministic predictions. Uncertainty is inherent in all aspects of hydrological modeling, including data uncertainty, model structural uncertainty, model parameter uncertainty, and prediction uncertainty. These uncertainties need to be characterized and quantified to ensure credible prediction, improve understanding of data limits and model deficiencies, and guide additional data collection and further model development in order to advance model predictability. In traditional, process-based hydrological modeling, significant efforts have been spent on uncertainty analysis (Vrugt et al., 2003; Pechlivanidis et al., 2011; Lu et al., 2012; sheng Zhan et al., 2013; Gan et al., 2014; Clark et al., 2016). Similar and even more extensive UQ efforts are required for ML simulation given its data-driven nature.

Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997), a ML model specifically designed for time-series prediction, can learn rainfall-runoff dynamic processes and hydrological system patterns from meteorological observations and streamflow data sequences. For example, when simulating daily streamflow, we use the previous several days of meteorological observations as inputs to predict streamflow on the current day. The observations contain noises/errors and this data uncertainty is propagated in the model learning and consequently affects streamflow predictions (Fang et al., 2020). Thus, it is important to understand how data quality influences ML model simulations and to quantify the confidence level of the prediction to assess trustworthiness. Additionally, the data-driven ML model usually produces reasonable predictions when the data in the unseen test period have similar features to those in the training period and can suffer from large extrapolation errors when the test data differ from the training set. In hydrological modeling, available training data are typically insufficient to accurately represent heterogeneous hydrological systems and the dynamics in these systems are often non-stationary due to climate change, land use/land cover change, extreme events, and environmental disturbances. As a result, it is likely that the trained ML model will encounter extrapolation issues when applied to new geographic regions and future climate projections. Therefore, it is crucial to identify whether the prediction results are reliable and if the trained model is suitable for the new condition.

UQ can help address the challenges of assessing the trustworthiness of ML model predictions and model reliability when applied to changing climate scenarios. For the training data, a well-calibrated UQ method can produce an uncertainty bound that precisely encloses a specified portion of the data consistent with the desired confidence level, e.g., for a 90% confidence interval, the uncertainty bound must cover about 90% of the training data to consider the data uncertainty and assess the model prediction's trustworthiness. For the unseen test data where the predicted values are not groundtruthed, a high-quality UQ method can produce increasing prediction uncertainty as the data move further away in both time and space from the training set, indicating that the ML model is outside of the training support and its prediction may not be trusted to avoid overconfidence. Hence, when we apply the trained model for prediction and UQ and compare the prediction interval width (PIW) of the unseen test data with that of the training data, we can infer the model's reliability and evaluate the prediction's trustworthiness. If the PIW of the test data is similar to that of the training data, it suggests that the test data are likely in-distribution (InD) and has similar features to the training data, and thus the trained ML model is reliable and suitable for the test set and its prediction can be trusted. The uncertainty bound additionally quantifies how trustworthy and likely the actual observations would be inside the bound to inform decision making. On the other hand, if the PIW of the test data is much larger than that of the training set, it suggests that the test data are out-of-distribution (OOD) and the trained model encounters something new that has not been learned before, so the ML model may fail to produce a reasonable, realistic prediction. UQ of ML model prediction is important when projecting the learned rainfall-runoff relationship to a new condition where groundtruthed data are unavailable. The quantified uncertainty can serve as a prediction error indicator to identify whether the trained model is reliable and how credible it is. In this way, UQ not only enables trustworthy prediction, but also allows hydrological modelers to know how ML model prediction accuracy may degrade and allows stakeholders to abstain from decisions due to low confidence.

However, UQ for ML model predictions is challenging and the development of a high-quality UQ method, which produces precise InD uncertainty and identifies OOD samples, is even more challenging. Generally speaking, there are two types of UQ-for-ML methods developed in the computational sciences community: prediction interval (PI) approaches which quantify uncertainty using intervals and non-PI approaches which quantify uncertainty using a distribution. The non-PI approaches can be further divided into Bayesian and non-Bayesian methods. Bayesian methods place priors on neural network (NN) weights and then infer predictive posterior distributions from the weights' distribution. The resulting posteriors are sensitive to the choice of the prior distributions. The Bayesian neural networks (BNNs) are usually solved by Markov Chain Monte Carlo sampling or some approximation methods such as variational inference (Lu et al., 2019) or Laplace approximation. BNNs have been criticized for slow training, overconfident predictions, and being computationally impractical for large-scale, deep-learning applications (Gal & Ghahramani, 2016a). Non-Bayesian methods include evidential learning that places priors directly over the likelihood function (Amini et al., 2020) and some ensemble methods that do not use priors such as deep ensembles (Lakshminarayanan et al., 2017), Monte Carlo dropout (Gal & Ghahramani, 2016b), and anchored ensembling (Pearce et al., 2020). Recently, some methods used deterministic deep learning for uncertainty estimation with some special NN architecture designs such as the spectral-normalized neural Gaussian process (J. Liu et al., 2020). These non-Bayesian methods usually involve a Gaussian assumption which might not be satisfied in hydrological applications where data noises are usually skewed and non-Gaussian (Schoups & Vrugt, 2010). These methods also may suffer from an overestimation of the uncertainty in training data caused by the symmetric uncertainty bound from the Gaussian assumption and result in an underestimation of the uncertainty in extrapolation (Zhang et al., 2021). Some of the non-PI methods have been applied in the hydrological modeling. For example, Zhu et al. (2020) combined Gaussian process with LSTM networks for probabilistic drought forecasting. Fang et al. (2020) used Monte Carlo dropout for soil moisture modeling and reported a tendency to underestimate uncertainty. Lu et al. (2021) also used Monte Carlo dropout to quantify streamflow predictive uncertainty in their application of LSTM networks for rainfall-runoff simulation. Recently, Klotz et al. (2022) established an uncertainty estimation benchmarking procedure and presented four ML baselines with one baseline being the Monte Carlo dropout.

The PI methods provide a lower and upper bound for a prediction such that the target falls between the bounds with a certain confidence level (e.g., 90%). PIs directly communicate uncertainty which provides understandable information for decision-making. Additionally, PI methods are computationally efficient and do not involve distributional assumptions, making them applicable to a wide range of scientific problems (Pearce et al., 2018a). Recently developed PI methods (Pearce et al., 2018a; Simhayev et al., 2020; Salem et al., 2020) tend to design sophisticated loss functions to obtain a well-calibrated interval. Although some studies have achieved promising results, their performance was sensitive to the unusual hyperparameters introduced into their customized loss functions. In practice, these hyperparameters usually require tedious fine tuning to achieve the desired performance, which makes these methods less practical and less robust when applied to hydrological applications. Some other PI methods, such as quantile regression approaches (Tagasovska & Lopez-Paz, 2019), could suffer from crossing issues where the calculated 90% prediction interval is even larger than the 95% interval (Zhou et al., 2020). Additionally, current PI methods usually lack the capability for OOD identification, making them less effective in indicating the model's reliability under the changing climate.

In this effort, we propose a PI method and integrate it with LSTM networks for improving streamflow predictability with UQ. The method is called PI3NN, which calculates prediction intervals based on three independent neural networks (Zhang et al., 2021; S. Liu et al., 2021, 2022). The first NN calculates the mean prediction, and the following two NNs produce the upper and lower bounds of the interval. After the three NNs' training, given a certain confidence level, PI3NN uses a root-finding algorithm to precisely determine the uncertainty bound that covers the desired portion of the data consistent with the confidence level. Additionally, PI3NN proposes a simple but effective initialization scheme for OOD identification. PI3NN is computationally efficient with three networks training; and for a different given confidence level, it just needs to perform the root finding step to calculate the shifting coefficients to precisely determine the corresponding interval and the calculated intervals do not suffer from the crossing issue. Additionally, PI3NN uses the standard mean squared loss and does not introduce extra hyperparameters, which enables a robust prediction performance and mitigates tedious parameter turning. Furthermore, PI3NN has an OOD identification capability which can produce a wider uncertainty for the predictions outside of the training data. Last but not the least, PI3NN is generalizable to various network structures and applicable to different data with no distributional assumptions, which makes it suitable for a wide range of ML-based hydrological applications.

In our previous work (S. Liu et al., 2021, 2022), we have integrated PI3NN to fully-connected, multilayer perceptron (MLP) networks and demonstrated its superior performance against several baselines using a range of diverse datasets. In this effort, we integrate our newly developed method with LSTM networks for streamflow prediction. LSTM has substantially different architectures from the MLP networks. In the implementation, we first separate the recurrent layers and the fully-connected dense layers of the LSTM network as two sets of networks. For the first recurrent network, we extract the temporal feature information from its outputs and use these outputs as the inputs for the second fully-connected network. Then, we perform PI3NN on this second fully-connected network and treat it as a MLP problem. This design improves training reliability, reduces the computational costs, and most importantly, reduces the requirement of large training data. We apply the proposed PI3NN-LSTM method for streamflow prediction and UQ to two diverse watersheds, the snow-dominant East River Watershed (ERW) in the western United States (US) and the rain-driven Walker Branch Watershed (WBW) in the southeastern US. We investigate the method's predictability of streamflow under different hydroclimatological conditions based on three components: prediction accuracy, quality and robustness of predictive uncertainty, and the OOD identification capability under a changing climate.

The major contributions of this study are:

- We develop a PI3NN method and integrate it into LSTM networks for improving streamflow prediction accuracy and credibility.
- PI3NN precisely quantifies the prediction uncertainty of the InD data with a desired confidence level and accurately identifies the OOD samples under a changing climate to avoid overconfident prediction.
- We demonstrate the PI3NN-LSTM model's prediction accuracy and UQ quality for streamflow predictions in both snow-dominant and rain-driven watersheds.

This paper is organized as follows. In Section 2, we describe the UQ method used for ML-based robust time-series prediction. In Section 3, we introduce the study watersheds and data used. Section 4 presents the results and discussion. Section 5 provides conclusions and recommendations for future research.

2 PI3NN method for UQ of ML model predictions

In this section, we introduce the PI3NN method to quantify ML model prediction uncertainty. We first describe the general procedure of PI3NN for a MLP dense network in a regression setting. Next, we discuss its capability of OOD identification. Lastly, we introduce the integration of PI3NN into the LSTM recurrent network for robust and credible time-series prediction.

2.1 Procedures of PI3NN for UQ

For a regression problem $y = f(\boldsymbol{x}) + \varepsilon$, we are interested in calculating the PIs to quantify the prediction uncertainty of the output y, where $\boldsymbol{x} \in \mathbb{R}^d$, $y \in \mathbb{R}$, and ε is the random noise with no distributional assumptions. In this study using ML models for daily streamflow prediction, $\boldsymbol{x}$ represents previous t days of meteorological observations; y represents the streamflow on the current day and ε denotes the data noise. The function f represents the LSTM network used to learn the rainfall-runoff relationship between $\boldsymbol{x}$ and y.

Based on a set of training data $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, PI3NN estimates predictions and quantifies predictive uncertainty using three networks and is implemented in three steps. Roughly speaking, PI3NN first trains three networks separately, where network $f_{\boldsymbol{\omega}}(\boldsymbol{x})$ is for mean prediction and networks $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ are for PI calculation. The PI3NN then uses root-finding methods to determine the upper bound $U(\boldsymbol{x})$ and lower bound $L(\boldsymbol{x})$ of the interval precisely for a given confidence level $\gamma \in [0, 1]$. Without a loss of generality, in the following we use basic MLP dense networks to explain the procedure and capability of PI3NN in Section 2.1 and 2.2 and then illustrate its integration into the recurrent network of LSTM in Section 2.3.

Step 1: train $f_{\boldsymbol{\omega}}(\boldsymbol{x})$ for mean prediction. This step follows a standard NN training for the deterministic prediction. The trained $f_{\boldsymbol{\omega}}(\boldsymbol{x})$ has two folds. First, the network outputs a mean prediction. Second, the differences (or residuals) between the prediction $f_{\boldsymbol{\omega}}(\boldsymbol{x})$ and the observation y will be used to construct the training set for networks $u_{\boldsymbol{\theta}}(\boldsymbol{x})$, $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ in the following Step 2.

Step 2: train $u_{\boldsymbol{\theta}}(\boldsymbol{x})$, $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ to quantify uncertainty. We first use the trained $f_{\boldsymbol{\omega}}(\boldsymbol{x})$ as a foundation to generate two positive data sets, $\mathcal{D}_{\text{upper}}$ and $\mathcal{D}_{\text{lower}}$, which include training data above and below $f_{\boldsymbol{\omega}}(\boldsymbol{x})$, respectively, i.e.,

$$\begin{aligned}\mathcal{D}_{\text{upper}} &= \big\{(\boldsymbol{x}_i, y_i - f_{\boldsymbol{\omega}}(\boldsymbol{x}_i)) \,\big|\, y_i \geq f_{\boldsymbol{\omega}}(\boldsymbol{x}_i), i = 1, \ldots, N\big\},\\ \mathcal{D}_{\text{lower}} &= \big\{(\boldsymbol{x}_i, f_{\boldsymbol{\omega}}(\boldsymbol{x}_i) - y_i) \,\big|\, y_i < f_{\boldsymbol{\omega}}(\boldsymbol{x}_i), i = 1, \ldots, N\big\}.\end{aligned} \tag{1}$$

Next, we use $\mathcal{D}_{\text{upper}}$ to train network $u_{\boldsymbol{\theta}}(\boldsymbol{x})$, and use $\mathcal{D}_{\text{lower}}$ to train network $l_{\boldsymbol{\xi}}(\boldsymbol{x})$. To ensure the outputs of $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ are positive, we add the operation $\sqrt{(\cdot)^2}$ to the output layer of both networks. The standard mean squared error (MSE) loss is used for training, i.e.,

$$\begin{aligned}\boldsymbol{\theta} &= \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{upper}}} (y_i - f_{\boldsymbol{\omega}}(\boldsymbol{x}_i) - u_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2,\\ \boldsymbol{\xi} &= \operatorname{argmin}_{\boldsymbol{\xi}} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{lower}}} (f_{\boldsymbol{\omega}}(\boldsymbol{x}_i) - y_i - l_{\boldsymbol{\xi}}(\boldsymbol{x}_i))^2.\end{aligned} \tag{2}$$

Step 3: construct the PI precisely via root-finding methods. The outputs of $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ approximate the positive and negative difference between the data and the prediction of $f_{\boldsymbol{\omega}}$, respectively. The bound defined by $[f_{\boldsymbol{\omega}} - l_{\boldsymbol{\xi}}, f_{\boldsymbol{\omega}} + u_{\boldsymbol{\theta}}]$ does not accurately quantify the PI. To calculate the interval that precisely encloses the desired portion of data consistent with the given confidence level, we additionally need to compute two coefficients α and β such that the upper and lower bounds defined below are a precise PI calculation.

$$\begin{aligned}U(\boldsymbol{x}) &= f_{\boldsymbol{\omega}}(\boldsymbol{x}) + \alpha u_{\boldsymbol{\theta}}(\boldsymbol{x}),\\ L(\boldsymbol{x}) &= f_{\boldsymbol{\omega}}(\boldsymbol{x}) - \beta l_{\boldsymbol{\xi}}(\boldsymbol{x}).\end{aligned} \tag{3}$$

For a given confidence level $\gamma \in [0, 1]$, we use the bisection method to determine the value of α and β by finding the roots of

$$Q_{\text{upper}}(\alpha) = 0, \qquad Q_{\text{lower}}(\beta) \quad = 0 \tag{4}$$

where

$$\begin{aligned}Q_{\text{upper}}(\alpha) &= \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{upper}}} \mathbf{1}_{y_i > U(\boldsymbol{x}_i)}(\boldsymbol{x}_i, y_i) - \frac{N(1-\gamma)}{2},\\ Q_{\text{lower}}(\beta) &= \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{lower}}} \mathbf{1}_{y_i < L(\boldsymbol{x}_i)}(\boldsymbol{x}_i, y_i) - \frac{N(1-\gamma)}{2}.\end{aligned} \tag{5}$$

In Eq. (5), N is the number of training data and $\mathbf{1}(\cdot)$ is the indicator function which counts how many training data points are outside the interval $[L(\boldsymbol{x}), U(\boldsymbol{x})]$. When this root-finding problem is solved, the number of training data falling in $[L(\boldsymbol{x}), U(\boldsymbol{x})] = [f_{\boldsymbol{\omega}} - \beta l_{\boldsymbol{\xi}}, f_{\boldsymbol{\omega}} + \alpha u_{\boldsymbol{\theta}}]$ will be exactly γN. Therefore, PI3NN produces an accurate uncertainty bound that precisely covers a specified portion of the data with a narrow-width interval. To make PI3NN work well, it is important to avoid overfitting in training $f_{\boldsymbol{\omega}}(\boldsymbol{x})$ in Step 1. An overfitted network may result in imbalanced data sizes of $\mathcal{D}_{\text{upper}}$ and $\mathcal{D}_{\text{lower}}$ and a possible unreliable training of $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$. The well-established regularization techniques such as L_1 and L_2 norm have been tested as a good penalty to avoid overfitting (Lu et al., 2021).

PI3NN is computationally efficient because it only requires three networks' training, and for a different given confidence level, it only needs to perform Step 3 to determine the corresponding PI without further training. The calculated intervals also do not suffer from the crossing issue. PI3NN is straightforward where the three networks are simple MLPs trained with a standard MSE loss. It does not introduce extra hyperparameters, unlike the customized loss in the modern PI methods (Pearce et al., 2018b; Simhayev et al., 2021). This enables a robust prediction performance and mitigates tedious hyperparameter turning. Additionally, PI3NN is generalizable to various network structures and applicable to different data with no distributional assumptions, which makes it suitable for a wide range of real-world applications. In Section 2.3, we integrate PI3NN to the LSTM network on time-series data for streamflow prediction.

2.2 OOD identification capability of PI3NN

A good-quality UQ method should not only produce a well-calibrated PI for the InD data to accurately quantify the uncertainty but also be able to identify the OOD samples to avoid overconfident predictions in the novel condition. In this section, we introduce the OOD identification capability of PI3NN. An OOD sample is defined as those data having a different distribution from or on the low probability region in the distribution of the training data. For example, if the training data come from a humid, warmer area, the prediction data in the arid, colder region which has dramatically distinct land covers could be the OOD samples. If the training set consists of data from wet years, the prediction data from dry years could be the OOD samples. As the OOD samples possess different features from the training set, it should be qualified with a large predictive uncertainty to show our low confidence when we use the trained model for extrapolation. The more it differs from the training data, the higher its predictive uncertainty would be. Thus, when we use the uncertainty to identify the OOD samples to indicate the ML model's reliability, the UQ method should be able to produce a larger prediction interval for the data further away from the training support.

PI3NN achieves OOD identification by properly initializing the output layer biases of networks $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$. Specifically, we add the following operations into the Step 2 before training $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$.

- Initialize the networks $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$ using the default option.
- Compute the mean outputs $\mu_{\text{upper}} = \sum_{i=1}^{N} u_{\boldsymbol{\theta}}(\boldsymbol{x}_i)/N$ and $\mu_{\text{lower}} = \sum_{i=1}^{N} l_{\boldsymbol{\xi}}(\boldsymbol{x}_i)/N$ using the training set.
- Modify the initialization of the output layer biases of $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$ to $c\,\mu_{\text{upper}}$ and $c\,\mu_{\text{lower}}$, where c is a relatively large number.
- Follow the Step 2 to train $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$.

Through above initialization strategy, outputs of networks $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ will be larger for the OOD samples than the InD data. Then after calculating the positive values of α and β in Step 3, it will correspondingly produce the larger uncertainty bounds $[L(\boldsymbol{x}), U(\boldsymbol{x})]$ for the OOD samples to indicate that their predictions are of low confidence.

The key ingredient in this OOD identification strategy is the modification of the biases of the network output layer. It is known that a MLP dense network is formulated as a piece-wise linear function. The weights and biases of hidden layers define how the input space is partitioned into a set of linear regions; the weights of the output layer determine how those linear regions are combined; and the biases of the output layer act as a shifting parameter. These network weights and biases are usually initialized with some standard distributions, e.g., uniform $\mathcal{U}[0,1]$ or Gaussian $\mathcal{N}[0,1]$, as default options. Setting the output layer biases to $c\mu_{\text{upper}}$ and $c\mu_{\text{lower}}$ with a large value of c will significantly lift up the initial outputs of $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$. During the training, the loss in Eq. (2) will encourage the decrease of $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ only for InD data (i.e., $\boldsymbol{x}_i \in \mathcal{D}_{\text{train}}$), not for OOD samples. Therefore, after training, $u_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $l_{\boldsymbol{\xi}}(\boldsymbol{x})$ will be larger in the OOD region than in the InD region (see Figure 1 in S. Liu et al. (2021) for an illustration). Correspondingly, the PIW of the OOD samples will be larger compared to that of the training data, based on which we identify the data/domain shift and indicate the extrapolation. Note that the exact value of c does not matter much, as long as it is a large positive value, e.g., we use $c = 100$ in this study. For training data, PI3NN will produce prediction intervals precisely enclosing $\gamma \times 100\%$ portion of data for a given confidence level $\gamma \in [0, 1]$ no matter how large the c values is, although a larger c in the network initialization may take a slightly longer training time for convergence. For the unseen test data, if they are InD with similar input features as the training set, PI3NN will produce uncertainty bounds with a similar width as the training data despite the large c value. If the test data are OOD outside of the training support, PI3NN will produce a larger PIW than that of the training data. The larger the c value is, the wider the PIW. Then, by comparing the PIWs of the test data with those of the training data, we diagnose whether the unseen test data are InD or OOD to quantify the trustworthiness of the ML model predictions. For OOD samples, we are not expected to accurately predict them, due to data-driven ML model deficiency, but more importantly it is to identify them to avoid overconfident predictions and provide a guidance for data collection to improve the predictability.

2.3 PI3NN-LSTM for robust time-series prediction

PI3NN can be generally applied to a wide range of network structures. It is straightforward for MLP networks to follow the above three steps in Section 2.1. In this section, we introduce its integration into the LSTM network for credible time series predictions. We first introduce the standard LSTM network, then describe how to use PI3NN to quantify its prediction uncertainty, and lastly depict the implementation of PI3NN-LSTM in steps.

LSTM is a special type of recurrent neural network to learn long-term dependence in time-series prediction, which makes it particularly suitable for daily streamflow simulation where lag times between precipitation (including both rainfall and snow) and discharge can be up to months. LSTM learns to map the inputs over time to an output, thus it knows what observations seen previously are relevant and how they are relevant for predictions enabling dynamic learning of temporal dependence. In daily streamflow modeling, the LSTM network reads previous t days of meteorological observations as inputs to predict streamflow on the current day. As shown in the bottom panel of Figure 1, each LSTM cell reads the input sequences $\boldsymbol{x}_t$ one time step at a time and the output from the previous time step is fed into the next cell as another input along with the input at current time step to affect the prediction, and so on. The outputs from the chain of LSTM cells are saved in the hidden states $\boldsymbol{h}_t$, which dynamically add, forget, and store information from the meteorological input sequences. Lastly, the LSTM network uses fully-connected dense layers to map the information in $\boldsymbol{h}_t$ to the quantity of interest y_t and predicts the current streamflow.

Essentially, the LSTM model consists of two subnets: a recurrent net and a MLP dense net. The recurrent subnet extracts input features and their temporal information and saves them in $\boldsymbol{h}_t$, i.e., $[\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_t] \rightarrow \boldsymbol{h}_t$. Subsequently, the dense subnet learns the input-output relationship from $\boldsymbol{h}_t$ to y_t, i.e., $\boldsymbol{h}_t \rightarrow y_t$. After the entire LSTM model is trained, the vector $\boldsymbol{h}_t$ saves all the information of the meteorological input sequences. Then, we can use $\boldsymbol{h}_t$ as a new set of inputs for the MLP network to predict the current streamflow of y_t and quantify its predictive uncertainty, without considering the recurrent subnet anymore. In this way, we successfully transform the UQ on the complex LSTM model into the UQ problem of the MLP network that we already know, which greatly simplifies the task.

To summarize, we perform the following three steps to integrate PI3NN into LSTM for time-series prediction and predictive uncertainty quantification (Figure 1) :

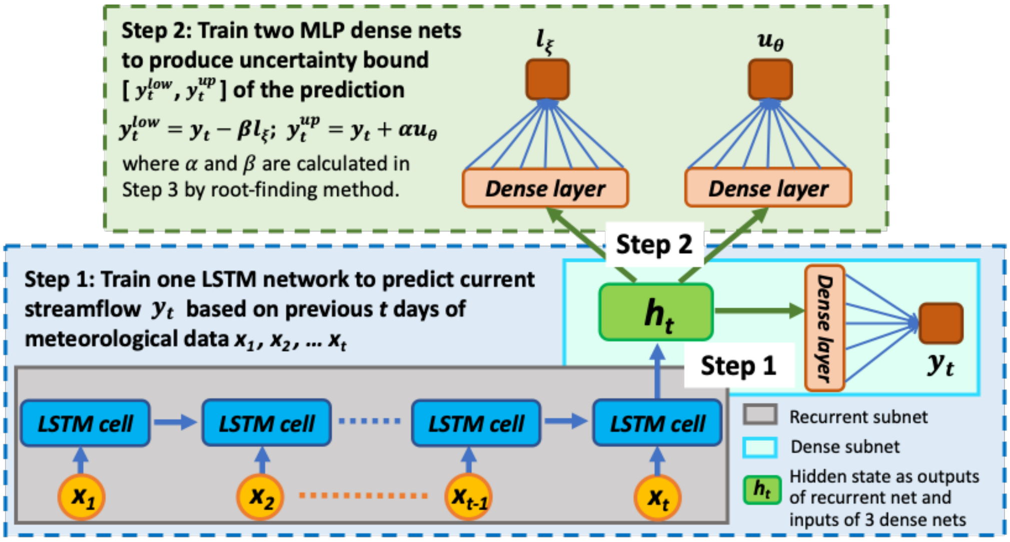

Figure 1. The workflow of the PI3NN-LSTM method where a LSTM network is trained for prediction and two MLP networks are trained for predictive uncertainty quantification.

- **Step 1** Train a LSTM model to predict y_t from multivariate input sequences of $[\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_t]$ in a standard way;
- **Step 2** Extract values of hidden state variable $\boldsymbol{h}_t$ as inputs and calculate the difference between the LSTM model prediction and observation on y_t as outputs to train two MLP dense nets for the PI calculation;
- **Step 3** Determine the PI of y_t precisely by computing the coefficients of α and β via the root-finding method.

In comparison to the three steps in Section 2.1, PI3NN-LSTM has the following similarities and differences. Step 1 is similar. Both train a ML model $f_{\boldsymbol{\omega}}(\boldsymbol{x})$, either a MLP model or a LSTM model, in a standard way for deterministic prediction. Step 2 is different, where the PI3NN-LSTM method here uses the hidden state variable $\boldsymbol{h}_t$ as the inputs to train the two MLP networks $u_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\xi}}$. The size of $\boldsymbol{h}_t$ is equal to the number of LSTM cells. Step 3 is the same as in Section 2.1. By employing the techniques in Section 2.2, the PI3NN-LSTM method can also examine the OOD samples in the time-series simulation and characterize the possible data/domain shift to avoid overconfident prediction. This strategy of network decomposition can be generally applied to other network structures. For example, we can decompose a convolutional neural network (CNN) model into a convolutional net and a MLP dense net, and decompose a graph neural network (GNN) model into a graph net and a MLP dense net. The recurrent net, convolutional net, and graph net in the LSTM, CNN, and GNN model, respectively, perform like an encoder which extracts temporal, spatial, and graph information into a hidden/latent variable. Then, we implement PI3NN on these latent variables to simplify the UQ task into the MLP problem. In this way, PI3NN can be applied for a variety of ML models in a computationally efficient and straightforward way.

3 Application of PI3NN to two diverse watersheds

We apply the PI3NN-LSTM method for daily streamflow prediction and UQ from meteorological observations in the snow-dominant East River Watershed (ERW) and the rain-driven Walker Branch Watershed (WBW) in the western and southeastern US, respectively. The two watersheds are distinctly different in their climatological patterns and hydrological dynamics. In the following, we first introduce the study area, data, and numerical experimental setup of each watershed and then we describe some prediction performance evaluation criteria.

3.1 Snow-dominant East River Watershed (ERW)

ERW is located in Colorado, US and it contains several headwater catchments in the Upper Colorado River basin. The watershed is about 300 km^2 and has an average elevation of 3266 m above mean sea level, with 1420 m of topographic relief and pronounced gradients in hydrology, vegetation, geology, and weather. The area is defined as having a continental, subarctic climate with long, cold winters and short, cool summers. The watershed has a mean annual temperature of 0°C, with average minimum and maximum temperatures of -9.2°C and 9.8°C, respectively; winter and growing seasons are distinct and greatly influence the hydrology. Annual average precipitation is approximately 1200 mm/yr and is mostly snow. River discharge is driven by snowmelt in late spring and early summer and by monsoonal-pattern rainfall in summer (Hubbard et al., 2018).

We consider data from two gauged stations, Quigley and Rock creek, both of which are headwater catchments with area of 576 acre and 800 acre, respectively. Each catchment includes four sequences of data: three input feature sequences of daily precipitation, maximum air temperature, and minimum air temperature, and one output sequence of daily streamflow. Quigley catchment has about two years of meteorological and streamflow observations from 09/01/2014 to 10/13/2016 with 774 daily measurements. Rock creek catchment has about three years of observations from 08/31/2014 to 10/04/2017 with 1131 daily measurements. In the LSTM simulation, we reserve the last year as unseen test data for prediction performance evaluation and use the remaining data for training. These two catchments have short records and it is a deliberate choice. As a new development of the PI3NN-LSTM method and the first application to the streamflow prediction, we want to first use a relatively small dataset for detailed analyses and deep understanding. And then in the second case study of the Walker Branch Watershed, we work on a long record of data.

Besides predicting streamflow, we also calculate its 90% prediction interval to quantify the predictive uncertainty. Additionally, we use PI3NN to investigate whether the unseen test data come from new climate conditions. If so, then the LSTM model predictions cannot be trusted and PI3NN should show a larger PIW compared to the training data. Specifically, for each catchment, following the procedure in Figure 1, we first use a standard LSTM model to predict streamflow from the meteorological observations. We then extract the hidden state information ($\boldsymbol{h}_t$) and construct two MLP dense networks to calculate the PI. In this calculation, we initialize the bias of the output layers of these two MLP dense nets with a large constant of c for the OOD detection (we investigate the influence of different c values on OOD identification capability in Section 4.1). In each network's learning, we perform a hyperparameter tuning using 20% of the training data. The network structures and the final hyperparameters used in the ERW simulations are listed below.

- For Quigley catchment: the LSTM network has a single recurrent layer with 128 nodes. The look-back window size is 45 days and the batch size is 64. Adam optimizer is used with a learning rate of 0.001. The two MLP dense nets for the PI calculation have a single layer with 10 nodes. To train the dense nets, we use the Adam optimizer with a learning rate of 0.001 and set the batch size to 32.
- For Rock creek catchment: the LSTM network has a single recurrent layer with 128 nodes. The look-back window size is 60 days and the batch size is 32. Adam optimizer is used with a learning rate of 0.001. The two MLP dense nets for the PI calculation have a single layer with 20 nodes. To train the dense nets, we use the Adam optimizer with a learning rate of 0.005 and set the batch size to 128.

In both catchments, log-transform of data is first applied and then the data are scaled to a range of $[-1, 1]$ for learning. Note that the above hyperparameters are standard for NNs. Our PI3NN method does not introduce extra hyperparameters which saves the effort of tedious tuning and more importantly promises reliable learning and stable prediction performance. Additionally, the dense networks used by PI3NN to quantify the LSTM prediction uncertainty have a simple structure which enables a data- and computationally-efficient training and UQ.

3.2 Rain-driven Walker Branch Watershed (WBW)

WBW is located in East Tennessee, US, and is part of the Clinch River which ultimately drains into the Mississippi River (Curlin & Nelson, n.d.; Griffiths & Mulholland, 2021). WBW includes the West Fork and East Fork catchments, which are 38.4 and 59.1 hectares in size, respectively. WBW has an average annual rainfall of 1350 mm and a mean annual temperature of 14.5 °C, which is consistent with a humid southern Appalachian region climate. The elevation ranges from 265 m to 351 m above mean sea level. Rain is the primary precipitation type in this region. Streamflow in both the West Fork and East Fork catchments is perennial and is fed by multiple springs (Johnson, 1989). We use data from the East Fork catchment in this study. The data consist of seven input sequences, including daily precipitation, maximum and minimum air temperature, maximum and minimum relative humidity, and maximum and minimum soil temperature, and one output sequence of daily streamflow. We have 14 years of observations from 01/01/1993 to 12/31/2006 with 5113 daily measurements. Given this long record of data, we reserve the last four years (2003-2006) as unseen test data for prediction performance evaluation and use the first ten years of data for training.

Similar to the ERW case study, we use the LSTM model to predict streamflow in the East Fork catchment of WBW, as well as use PI3NN to calculate its 90% prediction interval and to identify the possible OOD samples in the unseen test data. As WBW is a rain-driven watershed which has different meteorological and hydrological dynamics from the snow-dominant ERW, we used these contrasting watersheds to investigate whether PI3NN-LSTM is able to provide consistently good predictions under different conditions. Again in the East Fork catchment, we use 20% of the training data to determine the network structure and the hyperparameter values. The LSTM network has a single recurrent layer with 32 nodes. The look-back window size is 60 days and the batch size is 128. Adam optimizer is used with a learning rate of 0.001. The two MLP dense nets used by PI3NN to calculate the uncertainty have a single layer with 20 nodes, and the Adam optimizer with a learning rate of 0.005 is used for training with a batch size of 128.

3.3 Performance evaluation metrics

We use the Nash-Sutcliffe-Efficiency (NSE) to assess model prediction accuracy, and use the Prediction Interval Coverage Probability (PICP) and Prediction Interval Width (PIW) jointly to evaluate the quality of the UQ. NSE is an established measure used in the hydrological modeling to evaluate streamflow simulation accuracy based on the following equation:

$$NSE = 1 - \frac{\sum_{i=1}^{N}(y_i{}^{obs} - y_i{}^{pred})^2}{\sum_{i=1}^{N}(y_i{}^{obs} - \overline{y}^{obs})^2}, \tag{6}$$

where N is the total number of samples in evaluation, $y_i{}^{pred}$ represents predictions, $y_i{}^{obs}$ and $\overline{y}_i{}^{obs}$ are the observations and mean observations, respectively. The range of the NSE is (-inf, 1], where a value of 1 means a perfect simulation, a NSE of 0 means the simulation is as good as the mean of the observation and everything below zero means the simulation is worse compared to using the observed mean as a prediction. According to Moriasi et al. (2007), a NSE value greater than 0.50 is considered satisfactory, greater than 0.65 is considered good, and greater than 0.75 is very good.

PICP is defined as the ratio of samples that fall within their respective PIs. For example, for a sample set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, we use k_i to indicate whether the sample y_i is enclosed in its PI $[L, U]$, i.e.,

$$k_i = \begin{cases} 1, & \text{if } L(\boldsymbol{x}_i) \leq y_i \leq U(\boldsymbol{x}_i), \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Then, the total number of samples within upper and lower bounds is counted as:

$$s = \sum_{i=1}^{N} k_i. \tag{8}$$

Consequently, the PICP is calculated as:

$$PICP = \frac{s}{N} \times 100\%. \tag{9}$$

For each prediction data, the PIW is calculated as

$$PIW = U(\boldsymbol{x}) - L(\boldsymbol{x}) = \alpha u_{\boldsymbol{\theta}}(\boldsymbol{x}) + \beta l_{\boldsymbol{\xi}}(\boldsymbol{x}). \tag{10}$$

A high-quality UQ estimate should produce a PICP value close to its desired confidence level with a small PIW for InD data to demonstrate its accuracy and precision, and should be able to quantify uncertainty with a large PIW for the OOD data to avoid overconfident predictions.

4 Results and discussion

In this section, we evaluate the PI3NN-LSTM model's prediction performance. We assess the prediction accuracy using the NSE score and by comparing the observed and simulated hydrographs. We investigate PI3NN's UQ capability based on three aspects: the quality of the PI, the method's reliability, and data-, computational-efficiency, and its capability in identification of OOD samples. A well-calibrated UQ estimate should produce a reasonable, informative uncertainty bound quantifying the desired confidence level, e.g., for a 90% confidence level, the prediction interval should cover about 90% of the training data with a narrow width. Also, a high-quality UQ method should present an error-consistent uncertainty, i.e., for data points where the ML model has a low prediction accuracy, the method should yield a large uncertainty showing low confidence. Thus, when groundtruthed data are unavailable, it is reasonable to use the uncertainty bound as an error indicator to quantify the trustworthiness of the model prediction. Additionally, when we use the UQ method for applied issues (e.g., water resource management), we expect it to be reliable by involving only a few problem-dependent hyperparameters and being minimially constrained by the data distributional assumptions. Moreover, the method should be data-efficient given that hydrological observations can be sparse and expensive to obtain, and should be computationally efficient especially for large-scale and real-time water management applications. Last but not least, the UQ method should be able to detect the data/domain shift caused by the climate and environmental change to avoid overconfident predictions. In the following, we first analyze the results from the two snow-dominant catchments in ERW with short records of streamflow observations and then move to rain-driven WBW with a relatively long record of data. We discuss the results in ERW in detail and briefly summarize the findings in WBW as an extensive demonstration.

4.1 Streamflow prediction in snow-dominant ERW

Figure 2 depicts the two years of data in Quigley catchment where the top panel shows the one year of training data and the bottom panel shows the following year of unseen test data. This figure describes the rainfall-runoff dynamics of a typical snow-dominant watershed. Streamflow peaks in the spring/early summer and precipitation is highest in the winter from snowfall. The time lag between precipitation and streamflow can be explained by snow accumulation in the winter months and subsequent snow melt in spring. The LSTM network is able to successfully simulate this rainfall-runoff relationship and its memory effects by producing the predicted streamflow close to the observations based only on the precipitation and temperature inputs. The NSE value for the training data is 0.94 and for the test data is 0.87, suggesting a high prediction accuracy. Moreover, a closer look at the figure shows that in both training and test periods, the predicted hydrograph fits the general trends of the observation pretty well with a close peak flow timing and similar rising and falling limb shapes.

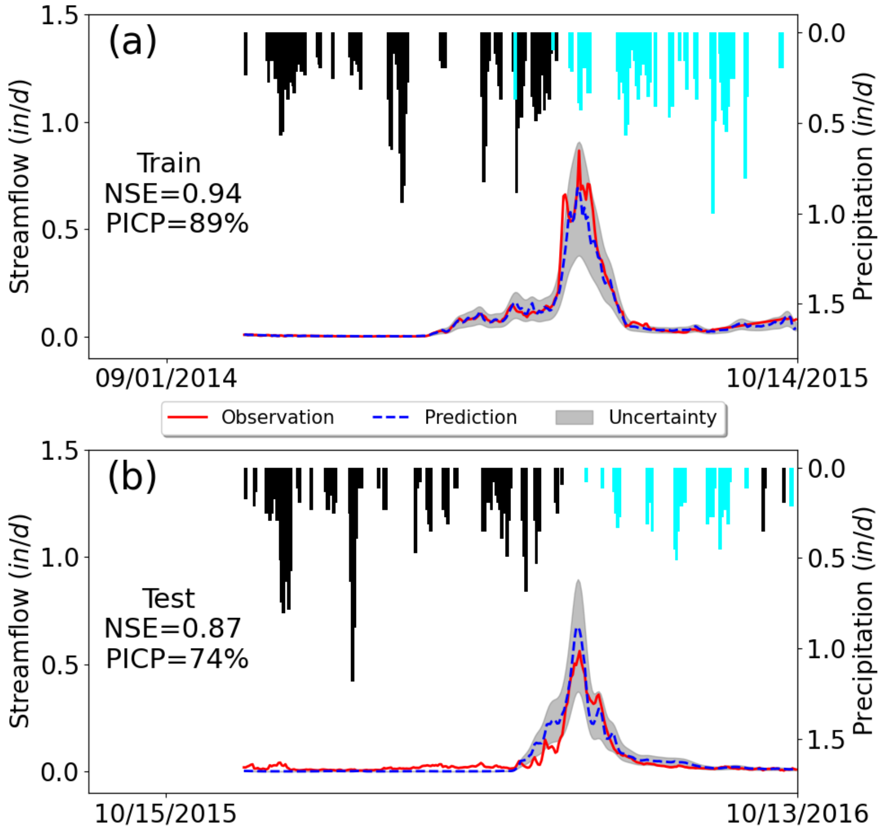

Figure 2. Predicted (dashed blue line) and observed streamflow (solid red line) in the snow-dominant Quigley catchment where the grey area quantifies the 90% predictive interval. Daily precipitation is plotted upside down on the top associated with the right y-axis, where snow (temperature below 0°C and in snow-water equivalents) is highlighted in black.

In Figure 2, we can also see that PI3NN accurately quantifies the prediction uncertainty where the PICP value of 89% in training data is close to its desired confidence level of 90%. Furthermore, the uncertainty bound covers the observations with a narrow width, demonstrating an informative UQ. Figure 3 summarizes the PIW for the training and test data using boxplots. It can be seen that the largest PIW in the training set of Quigley catchment is about 0.5 in/d, and it happens in simulating the peak flow where the LSTM model shows a relatively large error (Figure 2(a)). For the data points with accurate streamflow simulation, PI3NN produces a relatively narrow uncertainty bound with a small width interval, presenting realistically high confidence in line with the high accuracy. The similar PIW of the training and test data for Quigley shown in Figure 3 indicates that no OOD samples have been detected in this catchment and that the LSTM model predictions in the test period can be trusted. Indeed, we observe a high prediction accuracy of the test data as validated by the observations in Figure 2(b) and its PICP value suggests that about 74% of the test data are enclosed in the uncertainty bound.

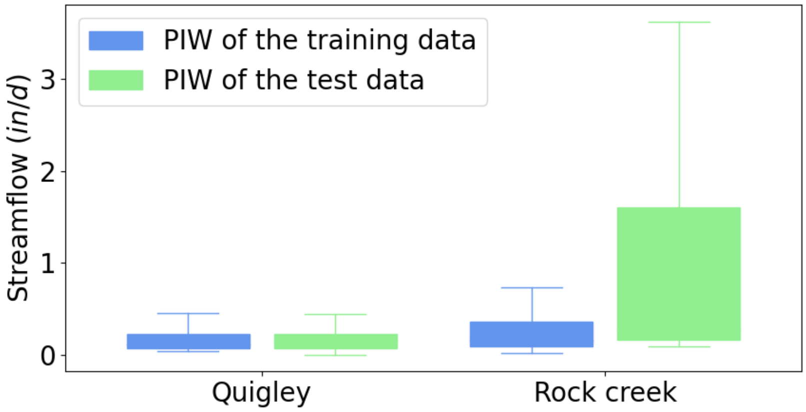

Figure 3. Prediction interval width (PIW) of the training and test data for the Quigley and Rock creek catchments in ERW. The similar PIW between the training and test data in Quigley indicates that the prediction for the test period can be trusted. In contrast, the largely different PIW between the training and test data in Rock creek suggests that its test period encounters some new climates that have not been seen before in training and the ML predictions may not be trusted.

This information is particularly useful in practice when the trained ML model is deployed for future projections or estimating the streamflow in ungauged catchments where no observation data are available. At this time, we need a prediction error indicator (which is usually calculated as the difference between the predictions and the observations) to indicate whether the ML model prediction can be trusted or not; after all, ML models are data driven and perform well when the unseen test data share similar properties with the training data. Hydrological dynamics are nonstationary due to multiple interacting drivers, such as climate change, land use, land cover, and other environmental changes. Without groundtruthed data, the uncertainty bound can serve as a prediction error indicator. When the PIW of the test data has a similar value to that of the training set, it suggests that the predictions can be trusted. When the PIW of the test data is much larger than the training data, this suggests that the model prediction accuracy is degrading and inferences should not be drawn from the predicted data due to the low prediction confidence. In Quigley catchment, we demonstrate that the training and test sets have similar PIWs and we further validate that the model predictions can be trusted by presenting a high consistency with the streamflow observations. Also, the calculated uncertainty bound encloses most of the actual data. Note that, we do not expect the 90% PI to enclose the exact 90% of the test data. PI3NN is guaranteed to produce the exact coverage for the training data because of its root-finding strategy. But for the unknown test data, a different feature from the training set would cause a different prediction performance and predictive uncertainty coverage.

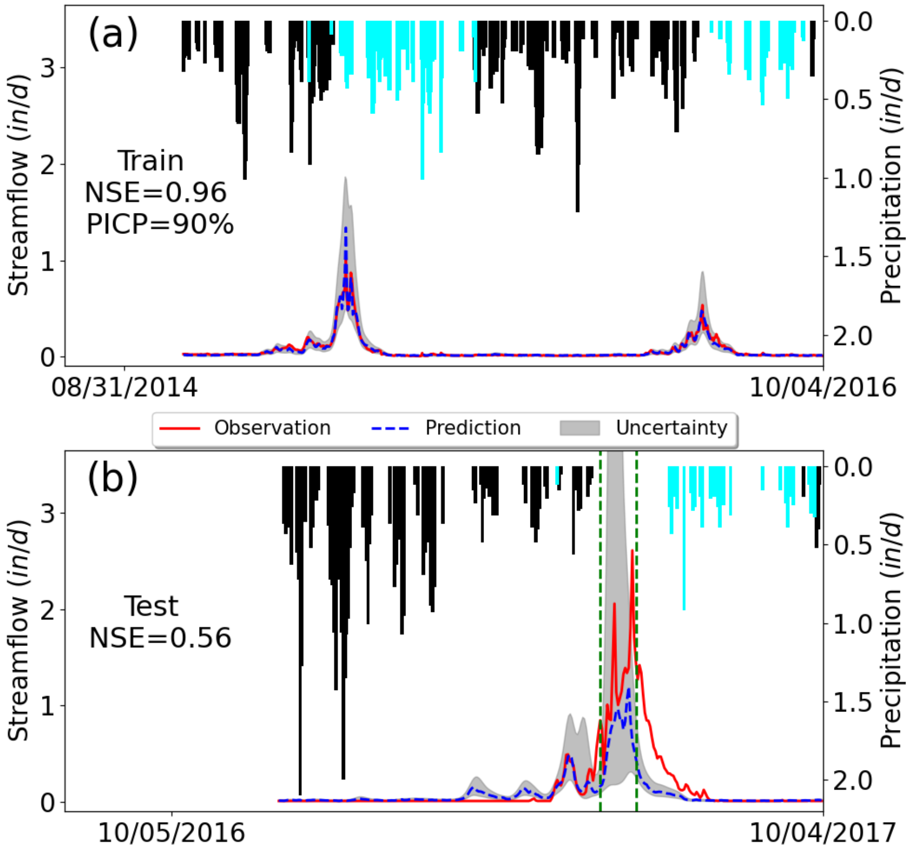

Figure 4. Predicted (dashed blue line) and observed streamflow (solid red line) in the snow-dominant Rock creek catchment where the grey area quantifies the predictive uncertainty. Corresponding daily precipitation is plotted upside down where snow (temperature below 0°C and in snow-water equivalents) is highlighted in black.

Figure 4 illustrates three years of data in Rock creek catchment where the top panel shows two years of training data and the bottom panel shows one year of test data. The test period of 2017 is a wet, cold year with unusually high precipitation (snow accumulation) in winter. Rock creek is a small headwater catchment and its streamflow is rather sensitive to the meteorological forcings, so the high precipitation in winter results in a correspondingly large peak flow in summer from snow melt, showing a data/domain shift relative to the training period of 2015-2016. In this case study, we want to investigate the LSTM model's capability in predicting the OOD samples caused by the new climate condition and more importantly to examine whether PI3NN can identify the data/domain shift and produce a large uncertainty by showing low confidence based on these anomalies.

Figure 3 clearly shows that the test data in Rock creek have a much larger PIW compared to the training set. This large difference in uncertainty bound indicates that the test samples contain some features that have not been learned before and they could fall outside of the training support. Thus, the model predictions cannot be trusted. Taking a close look at the hydrograph in the test period of Figure 4(b), we observe that the uncertainty bound in the peak flow regions between the two green dashed lines are remarkably high, and indeed this highly uncertain region has a larger prediction error where

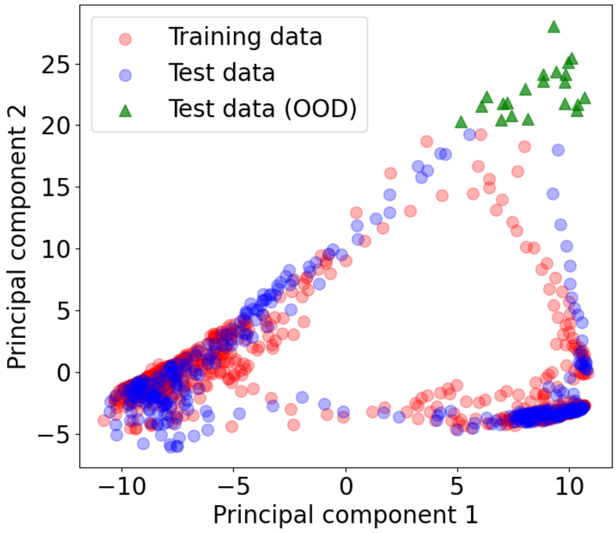

Figure 5. Projecting the training and test data of the input hidden state variable ($\boldsymbol{h}_t$) from its original 128 dimensions to the 2-dimensional space using principal component analysis for visualization. The 21 points (highlighted in green triangles) of the test data are identified as OOD samples, which suggests that their predicted streamflow cannot be trusted. These streamflow predictions are located between the two green dashed lines in Figure 4(b) which indeed shows poor prediction accuracy.

the model-predicted streamflow deviates from the observations the most. This underestimation of peak flow is understandable because the ML model only saw relatively low precipitation in the training period. Importantly, PI3NN is able to identify this underestimation by giving it a high uncertainty and a low confidence, suggesting that the model predictions on these data points should not be trusted, although the model has a good prediction performance in training. This information is very useful in real-world applications where groundtruthed data are unavailable. It can avoid overconfident predictions and guide reasonable decision making.

Note that, PI3NN identifies OOD samples based on their input features. If the data points are an anomaly in input space (e.g., extreme climates) then PI3NN can identify them and produce a high uncertainty in the output predictions (e.g., streamflow). However, if some data points have input features similar to the training set, although their predictions are poor, PI3NN or any other UQ methods cannot assign them large prediction uncertainties. In Rock creek catchment, the input space of the two MLP dense networks used for calculating the PIs are the 128 hidden states ($\boldsymbol{h}_t$). We project the training and test samples of $\boldsymbol{h}_t$ from their original 128-dimensional space to the 2-dimensional space using principal component analysis for visualization. Figure 5 indicates that there are 21 test data, at the upper right corner highlighted in green, relatively far away from other points and can be identified as OOD samples. We find that these 21 input data result in the streamflow predictions between the two green dashed lines in Figure 4(b) where PI3NN gives them large prediction uncertainties. This analysis explains the OOD identification capability of PI3NN. It demonstrates that if new climates make the trained ML model fail to accurately predict streamflow, PI3NN can correctly identify these new conditions and reasonably reflect their influence on streamflow prediction by producing a large uncertainty.

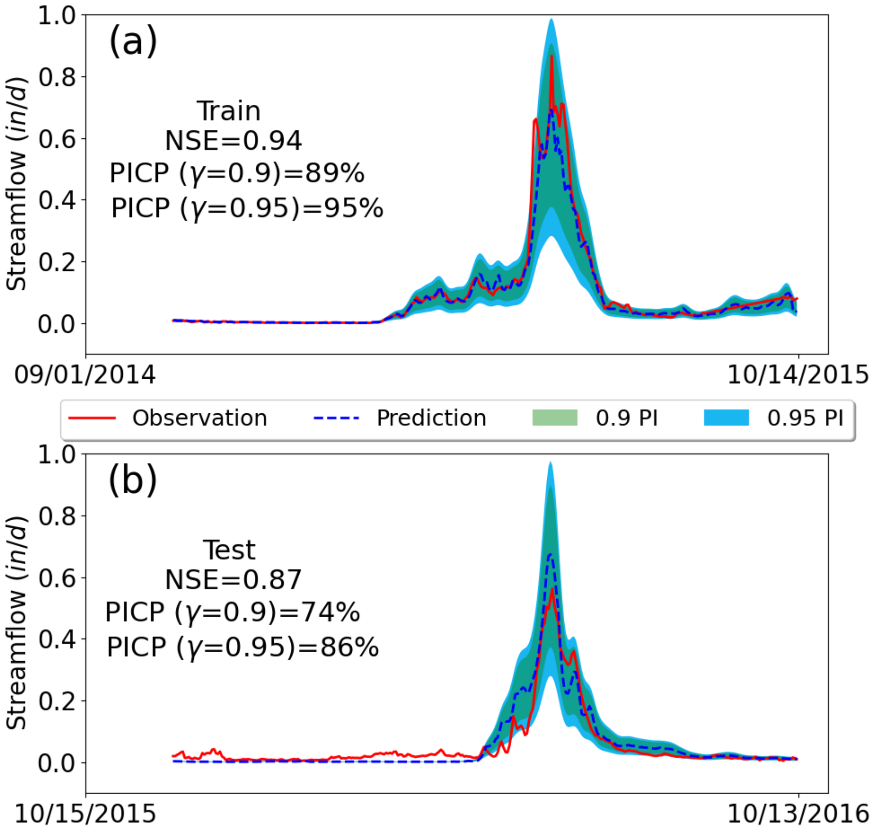

Figure 6. Streamflow observations and predictions for different confidence levels (γ) in the Quigley catchment. The 95% PI (γ=0.95) encloses 95% of the observations (PCIP=95%) and the 95% interval is wider than the 90% interval (γ=0.9) showing accuracy of the PI3NN method.

In the above analysis of ERW data, we demonstrate the PI3NN-LSTM's prediction accuracy, predictive uncertainty quality, and OOD identification capability. In the following, we discuss its reliability and efficiency. First of all, PI3NN is computationally efficient. It produces prediction intervals using three networks' training where the first network in this study is the standard LSTM for mean prediction, and the other two networks are MLP dense nets for UQ. In both catchments, we use a single-layer dense net whose training only takes 10-20 seconds and the computational cost of the following root-finding step is negligible (less than 1 second). Furthermore, for a different confidence level, PI3NN just needs to perform the root-finding step to determine the corresponding uncertainty bounds without further network training, and the calculated intervals are well-calibrated and do not suffer from the crossing issue. As illustrated in Figure 6 where both the 90% and 95% prediction intervals are plotted, the 95% PI encloses 95% of training data (PICP=95%) and its width is wider than the 90% interval (i.e., no crossing). Also, the 95% interval is able to cover more test data with a reasonably wider bound. Note that, this accurate calculation of PIs on streamflow predictions for a range of confidence levels only takes about 20 seconds of PI3NN after the standard LSTM model training. Besides, PI3NN is data efficient. Attributed to the LSTM network decomposition strategy (Section 2.3), we are able to use rather simple MLP dense nets to compute the uncertainty bound; and the simple network structures enable a small number of training data for an accurate learning. Here, by using one year of training data in Quigley and

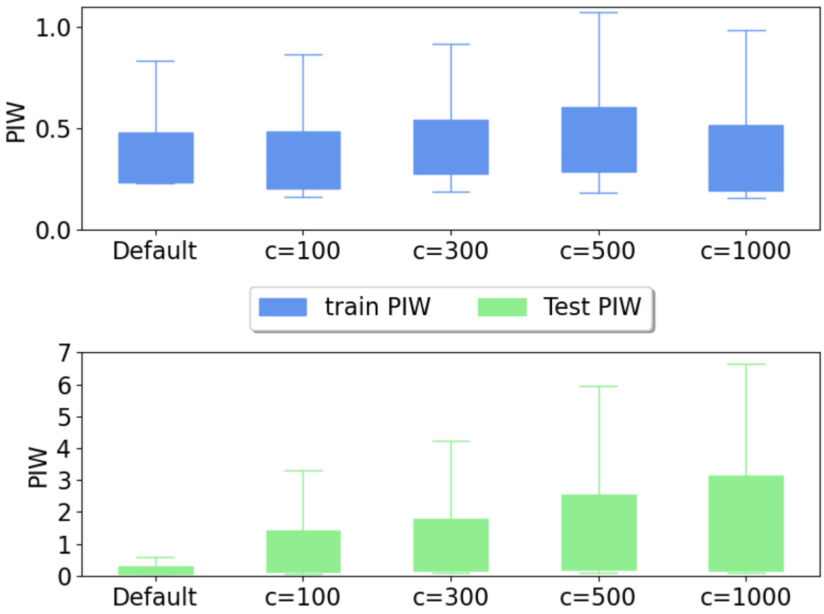

Figure 7. PIW of the training and test data for different output layer bias initialization in training the two interval networks for the Rock creek catchment. A larger c value initializes the bias to a larger value and the default c value usually draws a sample from a standard Gaussian distribution. Different c values do not affect training and any large c values here can identify the OOD samples with large PIWs, which indicates the reliability of PI3NN.

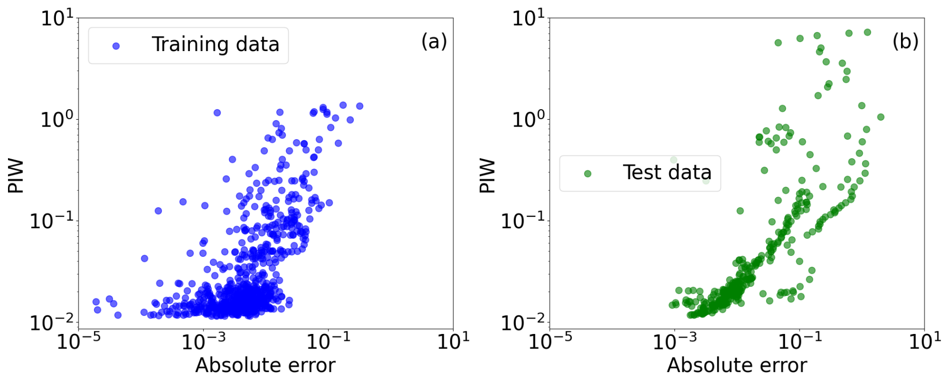

Figure 8. Scatter plots of absolute prediction errors VS. the PIW for both the training and test data sets in Rock creek catchment. The prediction interval shows error-consistent uncertainty where high uncertainties (i.e., large PIWs) correspond to large errors.

two years of training data in Rock creek, we are able to reasonably quantify the uncertainty and correctly identify the OOD samples.

Additionally, PI3NN is assumption-free and reliable. It does not involve a Gaussian assumption of the data noise, which makes it practically applicable to hydrological observations and able to generate an asymmetric uncertainty bound to precisely quantify the desired confidence level with a narrow width. Furthermore, PI3NN does not introduce extra hyperparameters allowing for reliable training and stable deployment in comparison to other state-of-the-art UQ methods. The only nonstandard parameter that needs to be specified in PI3NN is the constant c in initializing the output layer bias when using its OOD identification capability. In Figure 7, we demonstrate that as long as c is specified with a large positive value, PI3NN is able to detect the OOD samples by showing a larger PIW comparing to the training set. The exact value of c does not matter much and would barely affect the UQ quality. As we can see, with a different c, the PIWs of the training data are similar to each other and the specification of c does not affect the uncertainty coverage. For unseen test data, if OOD samples exists, a large c will lead to a large PIW enabling the identification of data/domain shift, although the larger the c value is, the more obvious the identification.

PI3NN is also a robust uncertainty estimate which produces error-consistent confidence. Figure 8 visualizes the relationship between absolute prediction errors and the PIW for both the training and test data sets in the Rock creek catchment. A clear monotonic trend is observed where the PIW increases as the increase of the errors, exhibiting decreasing confidence with the degradation of the prediction accuracy. Moreover, the identified OOD samples which cannot be accurately predicted by the ML model show a large PIW and a large error at the upper right corner of Figure 8(b). This error-consistent UQ property enables us to confidently use PI3NN as a ML model trustworthiness quantifier to diagnose when the model predictions can be trusted and when the results may fail, thus where to collect the data for the uncertainty reduction and the model prediction improvement.

4.2 Streamflow prediction in rain-driven WBW

In this section, we summarize the results from applying the PI3NN-LSTM model for streamflow prediction in rain-driven WBW and analyze the model's performance. Figure 9 depicts ten years of training (top) and four years of test data (bottom) in the East Fork of WBW. In comparison to Figures 2 and 4 that depict snow-dominant hydrological dynamics, this rain-driven watershed has many fewer snow days and shows a faster runoff response after a precipitation event. The training and test periods have similar magnitudes of precipitation on both annual and an event scale. In fact, we find that all the meteorological forcing inputs are of a similar magnitude in the training and test sets. PI3NN does not identify OOD samples in this dataset.

Figure 9 indicates that the LSTM network is able to simulate the streamflow reasonably well by showing a good fit to the observations. The overall NSE is 0.65 for the training data and 0.6 for the test data. Figure 10 plots each test year individually where both the predictive values and the 90% PI are depicted. Different years demonstrate different prediction accuracies, e.g., the NSE in 2005 is up to 0.78 while the subsequent year (2006) has a relatively low NSE of 0.50. In all the four test years, the LSTM model appears to underpredict peak flows, e.g., the observed peak flow is 617 L/s in 2003, but the predicted peak flow is 194 L/s; the observed peak flow is 274 L/s in 2004, and the predicted peak flow is 128 L/s. In this rain-driven watershed, peak flow happens during storms. It seems that the LSTM model has difficulties accurately predicting the magnitude of these event-triggered streamflows and the underprediction in peak flows results in the relatively low NSEs in most test years. Looking at the training period in Figure 9(a), it seems that even for the training data, LSTM has some underpredictions of the peak flows. To explore the reasons, we designed another numerical experiment where we used weighted mean squared errors as the loss function in training and the weight was proportional to the streamflow observations. Results indicate that the weighted mean squared error loss did not improve the underprediction of the peak flows. We think one possible reason is that these peak flows are erratic events which have relatively small observations compared to other streamflow data. ML models are data driven, and the small sets of data can deteriorate LSTM's capability in learning the underlying mechanism causing the high peak flows. Future investigations are needed to examine this possibility.

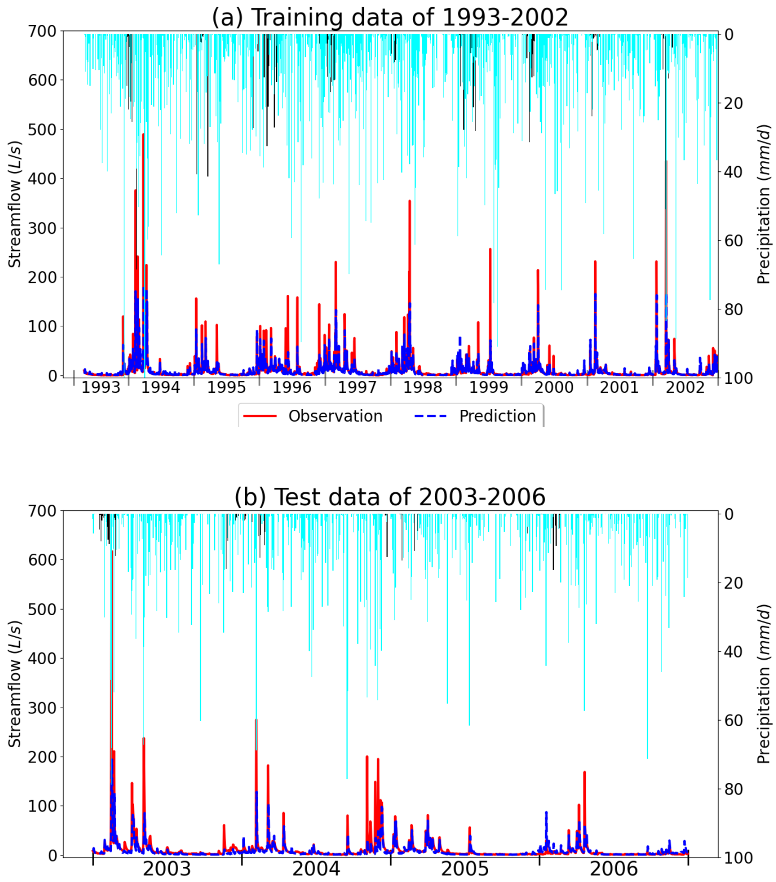

Figure 9. Predicted (dashed blue line) and observed streamflow (solid red line) in the East Fork of rain-driven WBW. Corresponding daily precipitation is plotted upside down on the top associated with the right y-axis, where snow (temperature below 0°C) is highlighted in black.

On the other hand, the peak flow timing in the test years is accurately predicted. For example, peak flow in 2003 was observed on the 47th day of the year and was predicted to occur on the 48th day. Peak flow was observed on the 37th day of 2004 and was predicted to happen on the 38th day. Both the observed and predicted peak flow happened on the 92nd day of 2005. Additionally, the LSTM model does a good job at predicting base flows. Zooming into the base flow regions by plotting the streamflow in logarithmic scale in Figure 10, we can see that the predicted base flows are close to the observations with a high consistency. Additionally, the predictive uncertainty in the test period can be precisely quantified by PI3NN, where the calculated PICP is close to the desired value of 90%. And most of the observed base flows are encompassed by the prediction intervals. PI3NN does not have a Gaussian distributional assumption on data so it can produce an asymmetric uncertainty bound to precisely cover the observations. For example, in August-October of 2003 where the model underpredicts streamflow, PI3NN produces a higher upper bound of the prediction interval to cover the observations. Note that, the predictive uncertainty associates with the prediction; if the predicted value greatly deviates from the observation and OOD samples are not detected, then we cannot expect the uncertainty bound encloses the observations. However, it is interesting to see that although the prediction accuracy is not very high for some years, e.g., the NSE is 0.5 in 2006, the prediction interval can cover the desired number of observations nicely with the PICP of 85%.

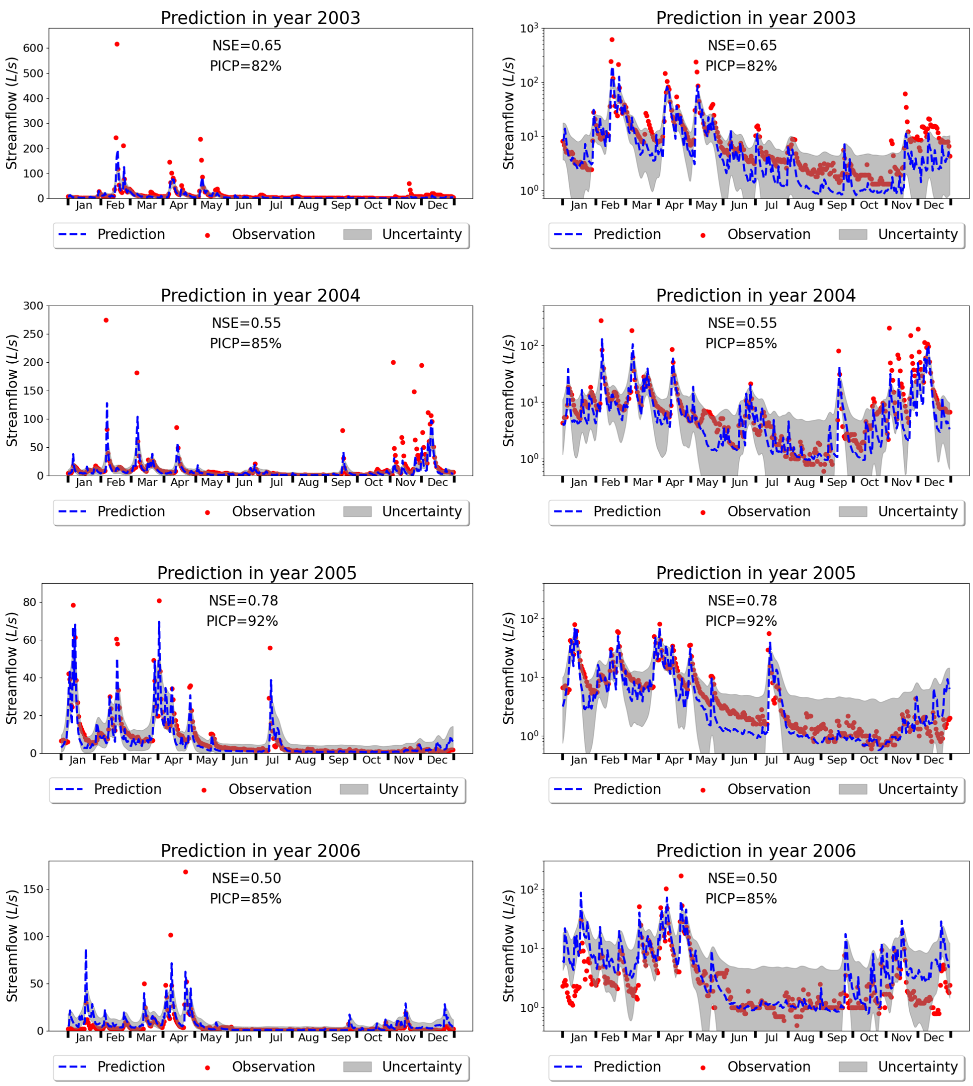

Figure 10. Predicted (dashed blue line) and observed streamflow (red dots) in the East Fork of rain-driven WBW where the grey area quantifies the 90% prediction interval. Figures in the left column have a linear scale on the y-axes to show the underprediction of peak flows while figures on the right have a logarithmic scale on the y-axes to show the accurate prediction and predictive uncertainty of base flows. Note that the y-axis range on each figure is different.

WBW has a complex geomorphological structure and interconnected hydrological processes (Griffiths & Mulholland, 2021). Many topographical, geological, soil, and ecological factors affect streamflow dynamics. However, in this model, we only consider a few meteorological variables as the inputs to simulate the streamflow, which may result in poor predictions due to the limited data and some missing information on important cause-effects. It is usually the case that the data, including the number of input variables and the number of observations, are too few to enable the ML model to accurately capture the underlying mechanisms of complex hydrological dynamics in watersheds. UQ cannot address the lack of data and it is not a replacement for data acquisition, but instead, it can guide cost-effective data collection. Additionally, it is promising to see here that the reasonably quantified uncertainty from PI3NN can encompass the desired number of observations despite the relatively poor fit.

5 Conclusions and future work

In this study, we propose a PI3NN method to quantify ML model prediction uncertainty and to integrate it with LSTM networks for streamflow prediction. Application of the PI3NN-LSTM method to both snow-dominant and rain-driven watersheds demonstrates its prediction accuracy, high-quality predictive uncertainty quantification, and the method's reliability, robustness, and both data- and computational-efficiency. For the test data which have similar features as the training data, PI3NN can precisely quantify prediction uncertainty with the desired confidence level; and for the OOD samples where the LSTM model fails to make accurate predictions, PI3NN can produce a reasonably large uncertainty indicating that the results are not trustworthy. Additionally, PI3NN produces error-consistent uncertainties where the prediction interval width increases as the prediction accuracy decreases. Therefore, when we apply the ML model to predict streamflow under future climate and at ungauged catchments where no groundtruthed data are available, the uncertainty quantifies the model predictions' trustworthiness, indicating whether the results should be trusted or further investigation needs to be conducted. PI3NN is computationally efficient, reliable in training, and generalizable to various network structures and data with no distributional assumptions. It can be broadly applied in ML-based hydrological simulations for credible predictions.

Although data are a key to improve ML model predictability, UQ is also crucial. From data we develop the data-driven ML model that is consistent with our knowledge, thus the model is more reliable under the changing climate and environmental conditions. On the other hand, UQ is significantly important for the trustworthiness of the predictions under these new conditions. Additionally, we can use UQ to guide the cost-effective data collection and to examine the model deficiency for further model development and improvement. In the future, we will apply PI3NN for streamflow prediction in multiple watersheds across the US and integrate it with different ML models for a variety of hydrological applications.

6 Data Availability Statement

The data for East River Watershed is available on ESS-DIVE (https://essdive.lbl.gov) and the data for Walker Branch Watershed can be downloaded from https://walkerbranch.ornl.gov. The PI3NN code is available at https://github.com/liusiyan/PI3NN.

7 Author Contributions

SL implemented the numerical experiments, summarized the results and prepared the figures. DL developed the algorithms, planned the research, plotted the figures, interpreted the results and drafted the manuscript. SLP, NAG, and EMP processed the data and interpreted the results. All the authors contributed to the manuscript writing.

Acknowledgments
This research was supported by the Artificial Intelligence Initiative as part of the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US DOE under contract DE-AC05-00OR22725. It is also sponsored by the ExaSheds project and the Critical Interfaces Science Focus Area project funded by the US DOE, Office of Biological and Environmental Research. We thank the late Pat Mulholland for collecting and maintaining the WBW streamflow data for many years.

References

Amini, A., Schwarting, W., Soleimany, A., & Rus, D. (2020). Deep evidential regression. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 14927–14937). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2020/file/aab085461de182608ee9f607f3f7d18f-Paper.pdf

Clark, M., Wilby, R., & Gutmann, E. e. a. (2016). Characterizing uncertainty of the hydrologic impacts of climate change. *Curr Clim Change Rep*, *2*, 55–64. Retrieved from https://doi.org/10.1007/s40641-016-0034-x

Curlin, J. W., & Nelson, D. J. (n.d.). Walker branch watershed project: Objectives, facilities, and ecological characteristics. Retrieved from https://www.osti.gov/biblio/4764827

Fang, K., Kifer, D., Lawson, K., & Shen, C. (2020, dec). Evaluating the potential and challenges of an uncertainty quantification method for long short-term memory models for soil moisture predictions. *Water Resources Research*, *56*(12). Retrieved from https://doi.org/10.1029%2F2020wr028095 doi: 10.1029/2020wr028095

Feng, D., Fang, K., & Shen, C. (2020). Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *Water Resources Research*, *56*(9), e2019WR026793. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR026793 (e2019WR026793 2019WR026793) doi: https://doi.org/10.1029/2019WR026793

Gal, Y., & Ghahramani, Z. (2016a, 20–22 Jun). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (Vol. 48, pp. 1050–1059). New York, New York, USA: PMLR. Retrieved from https://proceedings.mlr.press/v48/gal16.html

Gal, Y., & Ghahramani, Z. (2016b, 20–22 Jun). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (Vol. 48, pp. 1050–1059). New York, New York, USA: PMLR. Retrieved from https://proceedings.mlr.press/v48/gal16.html

Gan, Y., Duan, Q., Gong, W., Tong, C., Sun, Y., Chu, W., … Di, Z. (2014). A comprehensive evaluation of various sensitivity analysis methods: A case study with a hydrological model. *Environmental Modelling Software*, *51*, 269-285. Retrieved from https://www.sciencedirect.com/science/article/pii/S1364815213002338 doi: https://doi.org/10.1016/j.envsoft.2013.09.031

Griffiths, N. A., & Mulholland, P. J. (2021). Long-term hydrological, biogeochemical, and climatological data from walker branch watershed, east tennessee, usa. *Hydrological Processes*, *35*(3), e14110. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.14110 doi: https://doi.org/10.1002/hyp.14110

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hubbard, S. S., Williams, K. H., Agarwal, D., Banfield, J., Beller, H., Bouskill, N., … others (2018). The east river, colorado, watershed: A mountainous community testbed for improving predictive understanding of multiscale hydrological–biogeochemical dynamics. *Vadose Zone Journal*, *17*(1), 1–25.

Johnson, D. W. (1989). Site description. In D. W. Johnson & R. I. Van Hook (Eds.), *Analysis of biogeochemical cycling processes in walker branch watershed* (pp. 6–20). New York, NY: Springer New York. Retrieved from https://doi.org/10.1007/978-1-4612-3512-5_2 doi: 10.1007/978-1-4612-3512-5_2

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., … Nearing, G. (2022). Uncertainty estimation with deep learning for rainfall–runoff modeling. *Hydrology and Earth System Sciences*, *26*(6), 1673–1693. Retrieved from https://hess.copernicus.org/articles/26/1673/

2022/ doi: 10.5194/hess-26-1673-2022

Konapala, G., Kao, S.-C., Painter, S. L., & Lu, D. (2020). Machine learning assisted hybrid models can improve streamflow simulation in diverse catchments across the conterminous us. *Environmental Research Letters*, *15*(10), 104022.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, *22*(11), 6005–6022.

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. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR026065 doi: https://doi.org/10.1029/2019WR026065

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st international conference on neural information processing systems* (p. 64056416). Red Hook, NY, USA: Curran Associates Inc.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., & Lakshminarayanan, B. (2020). Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 7498–7512). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2020/file/543e83748234f7cbab21aa0ade66565f-Paper.pdf

Liu, S., Zhang, P., Lu, D., & Zhang, G. (2021). Pi3nn: Out-of-distribution-aware prediction intervals from three neural networks. *arXiv preprint arXiv:2108.02327*.

Liu, S., Zhang, P., Lu, D., & Zhang, G. (2022). PI3NN: Out-of-distribution-aware prediction intervals from three neural networks. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=NoB8YgRuoFU

Lu, D., Konapala, G., Painter, S. L., Kao, S.-C., & Gangrade, S. (2021). Streamflow simulation in data-scarce basins using bayesian and physics-informed machine learning models. *Journal of Hydrometeorology*, *22*(6), 1421–1438.

Lu, D., Liu, S., & Ricciuto, D. (2019). An efficient bayesian method for advancing the application of deep learning in earth science. In *2019 international conference on data mining workshops (icdmw)* (pp. 270–278).

Lu, D., Ye, M., & Hill, M. C. (2012). Analysis of regression confidence intervals and bayesian credible intervals for uncertainty quantification. *Water Resources Research*, *48*(9). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011WR011289 doi: https://doi.org/10.1029/2011WR011289

Moriasi, D. N., J. G., A., M. W., V. L., R. L., B., R. D., H., & T. L., V. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*.

Pearce, T., Brintrup, A., Zaki, M., & Neely, A. (2018a, 10–15 Jul). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 4075–4084). PMLR. Retrieved from https://proceedings.mlr.press/v80/pearce18a.html

Pearce, T., Brintrup, A., Zaki, M., & Neely, A. (2018b, 10–15 Jul). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 4075–4084). PMLR. Retrieved from http://proceedings.mlr.press/v80/pearce18a.html

Pearce, T., Leibfried, F., & Brintrup, A. (2020, 26–28 Aug). Uncertainty in neural networks: Approximately bayesian ensembling. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the twenty third international conference on artificial intelligence and statistics* (Vol. 108, pp. 234–244). PMLR. Retrieved from https://proceedings.mlr.press/v108/pearce20a.html

Pechlivanidis, I., Jackson, B., Mcintyre, N., Wheater, H., et al. (2011). Catchment scale hydrological modelling: A review of model types, calibration approaches and uncertainty analysis methods in the context of recent developments in technology and applications.

Rasouli, K., Hsieh, W. W., & Cannon, A. J. (2012). Daily streamflow forecasting by machine learning methods with weather and climate inputs. *Journal of Hydrology*, *414-415*, 284-293. Retrieved from https://www.sciencedirect.com/science/article/pii/S0022169411007633 doi: https://doi.org/10.1016/j.jhydrol.2011.10.039

Salem, T. S., Langseth, H., & Ramampiaro, H. (2020). Prediction intervals: Split normal mixture from quality-driven deep ensembles. In *Conference on uncertainty in artificial intelligence* (pp. 1179–1187).

Schoups, G., & Vrugt, J. A. (2010). A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic, and non-gaussian errors. *Water Resources Research*, *46*(10). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2009WR008933 doi: https://doi.org/10.1029/2009WR008933

Shamshirband, S., Hashemi, S., Salimi, H., Samadianfard, S., Asadi, E., Shadkani, S., … Chau, K.-W. (2020). Predicting standardized streamflow index for hydrological drought using machine learning models. *Engineering Applications of Computational Fluid Mechanics*, *14*(1), 339-350. Retrieved from https://doi.org/10.1080/19942060.2020.1715844 doi: 10.1080/19942060.2020.1715844

sheng Zhan, C., meng Song, X., Xia, J., & Tong, C. (2013). An efficient integrated approach for global sensitivity analysis of hydrological model parameters. *Environmental Modelling Software*, *41*, 39-52. Retrieved from https://www.sciencedirect.com/science/article/pii/S1364815212002563 doi: https://doi.org/10.1016/j.envsoft.2012.10.009

Shortridge, J. E., Guikema, S. D., & Zaitchik, B. F. (2016). Machine learning methods for empirical streamflow simulation: a comparison of model accuracy, interpretability, and uncertainty in seasonal watersheds. *Hydrology and Earth System Sciences*, *20*(7), 2611–2628. Retrieved from https://hess.copernicus.org/articles/20/2611/2016/ doi: 10.5194/hess-20-2611-2016

Simhayev, E., Katz, G., & Rokach, L. (2020). Piven: A deep neural network for prediction intervals with specific value prediction. *arXiv preprint arXiv:2006.05139*.

Simhayev, E., Katz, G., & Rokach, L. (2021). *Piven: A deep neural network for prediction intervals with specific value prediction.*

Tagasovska, N., & Lopez-Paz, D. (2019). Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, *32*.

Tongal, H., & Booij, M. J. (2018). Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. *Journal of Hydrology*, *564*, 266-282. Retrieved from https://www.sciencedirect.com/science/article/pii/S0022169418305092 doi: https://doi.org/10.1016/j.jhydrol.2018.07.004

Vrugt, J. A., Gupta, H. V., Bouten, W., & Sorooshian, S. (2003). A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resources Research*, *39*(8). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002WR001642 doi: https://doi.org/10.1029/2002WR001642

Xu, T., & Liang, F. (2021). Machine learning for hydrologic sciences: An introductory overview. *WIREs Water*, *8*(5), e1533. Retrieved from `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1533` doi: https://doi.org/10.1002/wat2.1533

Zhang, P., Liu, S., Lu, D., Zhang, G., & Sankaran, R. (2021). *A prediction interval method for uncertainty quantification of regression models* (Tech. Rep.). Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Zhou, F., Wang, J., & Feng, X. (2020). Non-crossing quantile regression for distributional reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 15909–15919). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2020/file/b6f8dc086b2d60c5856e4ff517060392-Paper.pdf`

Zhu, S., Luo, X., Yuan, X., & Xu, Z. (2020). An improved long short-term memory network for streamflow forecasting in the upper yangtze river. *Stochastic Environmental Research and Risk Assessment*, *34*(9), 1313–1329.