# Final Report [A Web Platform for Dynamical Streamflow Prediction Using Machine Learning and Deep Learning Methods]

**Sadegh Sadeghi Tabas**
sadeghs@clemson.edu

**Nushrat Humaira**
nhumair@g.clemson.edu

**Pawan Madanan**
pmadana@g.clemson.edu

**Siddish P Rao**
siddisr@g.clemson.edu

**Meghan Patil**
mmpatil@g.clemson.edu

## Abstract

Runoff prediction from meteorological observations provides the basic information for the management of water resources, the design of hydro-power plants and the planning of irrigation schemes. They also act as a backbone in reducing the damages and casualties from floods, which are among the most frequent and destructive natural hazards. Various approaches exist, ranging from physically based over conceptual to fully data-driven models. In this research a number of data driven (machine learning and deep learning) and data mining methods including multi-layer perceptron (MLP), long short-term memory (LSTM) and a hybrid deep learning method of convolutional neural network and LSTM have been implemented in a web designed platform to predict sequential flow rate values based on a set of collected runoff factors in a global scale (North America, South America and Africa). This study has focused on the effects of input data characteristics on model performance (sequential data), therefore, the type of input data, the amount of input data, and the correlation of the data series has been considered. The datasets are gathered from different databases including CAMELS, NCDC and GRDC. In order to remove or impute the missing values and preprocess the available data, several methods have been used and a reliable dataset was generated to be used in the modeling part of the research. The developed prediction models were validated and tested using NSE, KGE and TRMSE criteria. The findings of this study suggest the potential of applying data-driven models in the field of hydrological runoff prediction.

## 1 Introduction

### 1.1 Problem Specification

Watershed simulations help understand the past and current state of rainfall-runoff processes in basins and provide a way to explore the implications of management and planning decisions and imposed changes (such as land use change, climate change). In complex environmental systems such as the coastal plain watersheds, very significant modeling efforts have gone through lumped and physical based simulations (Sadeghi-Tabas et al., 2017; Samadi et al., 2020, 2017; Samadi and Meadows, 2017). Indeed, in pursuit of improved accuracy, there is a plethora of hydrological tools that have been enhanced and implemented for complex rainfall-runoff simulations. In the realm of process-based hydrologic modeling, time-series machine learning (ML) or data-driven algorithms have been recently utilized to improve short- and long-term streamflow predictions at various scales

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

and domains (e.g., Fang et al., 2017; Kratzert et al., 2018; Shen, 2018; Shen et al., 2018; Wu et al., 2018).

Data-driven algorithms attempt to estimate the mapping function (f) from the input variables (x) to numerical or continuous output variables (y) and to understand time-series natural temporal ordering across long-term records. These methods can directly learn patterns hidden in time-series data, without requiring manually-designed features or making strong physical assumptions (Kasiviswanathan et al., 2016; LeCun et al., 2015; Schmidhuber, 2015) and are good at handling large datasets with high dimensionality and heterogeneous feature types. Many studies have demonstrated that ML models can outperform other state-of-the-art techniques in hydrologic simulation (Yang et al., 2020). Among various ML algorithms, regression-based models such as Support Vector Machines (SVM), Random Forest (RF), Multi-layer Perceptron (MLP), and Decision Tree models have been widely applied in streamflow simulations and forecasting.

### 1.2 Related Studies

To give a few selective examples, Sivapragasam and Muttil, (2005) studied the application of SVM to extend the development of rating curves at three gauging stations in Washington, USA. Their results indicated that SVM was better suited for rating curves extrapolation compared to widely used logarithmic method and higher order polynomial fitting method. Sadler et al., (2018) applied RF methods for the coastal urban stormwater prediction in Virginia, USA. They used quality-controlled, crowd-sourced street flooding reports ranging from 1 to 159 per storm event for 45 storm events to train and evaluate RF models. Their results showed that RF performed better than Poisson regression at predicting the number of flood reports and had a lower false negative rate. Bui et al., (2016) proposed a new artificial intelligence approach based on neural fuzzy inference system and metaheuristic optimization for flood susceptibility modeling (MONF) over the Tuong Duong district in Central Vietnam. They found that MONF outperformed other machine learning algorithms including Multi-layer Perceptron (MLP) and Decision Tree for flood susceptibility mapping.

Despite the expanded use of ML and DL models in streamflow simulation, few studies have used advanced data-driven methods to model streamflow within coastal environments. The closest work may be the statistical analysis of streamflow records in the United States to compute the probabilities of high and low flow events in the past several decades along with the projected changes in the coming decades (Asadieh and Krakauer, 2017; Campbell et al., 2011; Hidalgo et al., 2009).

### 1.3 Motivation and Novelty

This study proposes a novel web platform to predict daily streamflow in a global scale (North America, South America and Africa) using various state-of-the-art ML and deep learning (DL) models as data-driven approaches. Data-driven approaches can be appropriate tools for streamflow simulation due to the complexity of modeling rainfall-runoff processes, which makes using a physical model difficult. This study investigated and compared three different ML and DL algorithms including MLP, LSTM and a hybrid method of LSTM and Convolutional Neural Network (CNN). Proposed modeling approach is divided into multiple stages. First stage performs preprocessing of the available datasets and has been done in order to impute the missing values with an accurate estimate using different methods. In the second stage, all data driven models are trained and saved for inference in the next stage. Final stage is web application with all models and datasets deployed. Web application demonstrates streamflow prediction for test time period across all location visualized in a global map. Application offers the user to select a model to test the time series prediction.

## 2 Methodology

This section contains four different parts. In the first part different case studies have been introduced and it followed by the second part in which the data driven methods implemented in this study has been introduced. The third part describes the web design platform proposed in this research followed by the last part which introduces the various performance assessment criteria which have been used to validate the results.

### 2.1 Case Studies and Data

#### 2.1.1 North America

The underlying data for North America case study is the CAMELS data set (Addor et al., 2017b; Newman et al., 2014). The acronym stands for "Catchment Attributes for Large-Sample Studies" and it is a freely available data set of 671 catchments with minimal human disturbances across the contiguous United States (CONUS). The data set contains catchment aggregated (lumped) meteorological forcing data and observed discharge at the daily timescale starting (for most catchments) from 1980. The meteorological data are calculated from three different gridded data sources (Daymet, Thornton et al., 2012; Maurer, Maurer et al., 2002; and NLDAS, Xia et al., 2012) and consists of day length, precipitation, shortwave downward radiation, maximum and minimum temperature, snow-water equivalent and humidity. We used the Daymet data, since it has the highest spatial resolution (1 km grid compared to 12 km grid for Maurer and NLDAS) as a basis for calculating the catchment averages and in order to have same inputs for the entire case studies in this research precipitation and antecedent day streamflow considered as input to each data driven method. In this research, we have selected 18 basins as sample watersheds of the North America region.

#### 2.1.2 South America

The data for the South America case study was acquired from two different sources, namely GRDC (Global Runoff Data Center) and NCDC (National Climatic Data Center). The GRDC dataset contains streamflow data collected from 10,063 stations across the globe, of which we selected stations within South America. The NCDC dataset, which is maintained by the NCEI (National Center for Environmental Information), contains precipitation, wind speed and air temperature data collected from various stations globally of which we selected stations within South America. The wind speed and air temperature in this dataset is sparse and most stations only contain precipitation data which is what we have considered. We found paired sets of GRDC and NCDC stations, with suitable geographic proximity, using the precipitation and antecedent day streamflow data as inputs for predicting streamflow at time step t in GRDC stations. We selected 8 such pairs as an example for our research. Next the missing values were imputed and the data was used to train and test the models employed in this study.

#### 2.1.3 Africa

Similar to South America case study, we used GRDC and NCDC datasets for Africa as well. As was the case with South America, the wind speed and air temperature components of the NCDC dataset were sparse and thus we selected precipitation data while we used GRDC station dataset for streamflow data. 6 pairs of GRDC and NCDC stations were selected, with sufficient proximity (both located in same watershed) for our research. Next the missing values were imputed and the data was used to train and test the models employed in this study.

### 2.2 Data Driven Models

#### 2.2.1 Multi-Layer Perceptron (MLP)

Multi-layer perceptron is a class of feedforward artificial neural networks (ANN) with input/output layers and several hidden layers. Nonlinear activation functions are used in the neurons to extract, learn, and remember the nonlinear features and sub features from the input data. Backpropagation is a family of methods which is always used to update the parameters in the ANN by calculating the gradient of a loss function with respect to all the parameters and back propagating the training errors. MLP consists of perceptron, or neuron that has four sections: (i) input values, (ii) weights and bias, (iii) weighted sum, and (iv) activation function. In a multilayer perceptron, the output of one layer's perceptron is the input of the next layer. The output of the final perceptron, in the "output layer", is the final prediction of the perceptron learning model. For the MLP simulation, the rectified linear unit (ReLU) transfer function was incorporated into the neurons of the hidden layer and output layer, respectively. The number of hidden neurons was identified by trial and error procedure which started with one hidden neuron initially and increased to fifty with a step size of one at each trial and finally the optimum value of fifteen hidden layers obtained. For each set of hidden neurons, the network was trained to minimize the Mean Squared Errors (MSE) at the output layer. Levenberg-Marquardt

algorithm was used to update the values of weights and biases of the MLP simulation. The model run for 10,000 iterations and the training was stopped when there was no significant improvement in the performance. The parsimonious structure that resulted in minimum error and maximum efficiency during training was selected as the final form of MLP.

#### 2.2.2 Long Short-Term Memory (LSTM)

For every region, we have collected observations on precipitation and streamflow. Since we have more than one observation for each time step, our model can be categorized as multivariate time series prediction model. Thus, we have two or parallel input time series and an output time series for streamflow that is dependent on the input time series. An LSTM model needs contextual information to learn a mapping from an input sequence to an output value. For this purpose, we have chosen to include streamflow values from previous time step as input to the model. Time series were split into 70:30 ratio for training and testing data. Both training and testing data were normalized using standard scalar. All inputs to the model were shaped into the form [batch size, timesteps, features count]. LSTM model had an input layer, followed by LSTM layer of 30 units and ReLU activation layer. Final output layer of one node produces the regression score defined as coefficient of determination implemented by scikit-learn package. Original observation streamflow data and predicted simulated data were saved for each station to calculate the performance as described in later sections.

#### 2.2.3 Hybrid Model (CNN-LSTM)

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. A CNN requires much less pre-processing as compared to other classification algorithms. Through enough training, CNNs have the ability to learn filters/characteristics on their own. Though it is worth noting that in primitive methods filters are hand-engineered. A CNN is best known for performance with two-dimensional image data. They are regularized version of MLP. CNN can also extract and learn features from one-dimensional sequence data such as one of the series from a multivariate multi input time series. At the core of any CNN is it's convolution layer. The convolution layer has several parameters which also consist of a set of learnable filters, these are also called kernels. The filters extend through the full depth of the input volume, though they have a small receptive field. Each filter is convolved across the width and height of the input volume during the forward pass. It computes the dot product between the entries of the filter and the input during this time and produces a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input. We used a hybrid model with CNN in the lower layer end followed by LSTM layer and fully connected dense layer on the output. We used 1 dimensional convolutional layer for our model and ReLU activation for both convolution and LSTM layer.

### 2.3 Web Platform

The main web application was done in HTML and JavaScript with the help of Leaflet (JavaScript library for interactive maps). This library provides interactive display of geographic information. We also used JavaScript to implement certain UI elements like popups and legend for the layers in the application. We provided option to select the model among the different model that we used, for which the corresponding model prediction data will be shown. A basic preview of the proposed web app is presented in the Figures 1 for North America, South America and Africa respectively. For more information all the the package (codes), datasets and the modeling results are uploaded to this Github repository which is publicly available [1].

[1] `https://github.com/sadeghitabas/CPSC8810-Mining-Massive-Data`

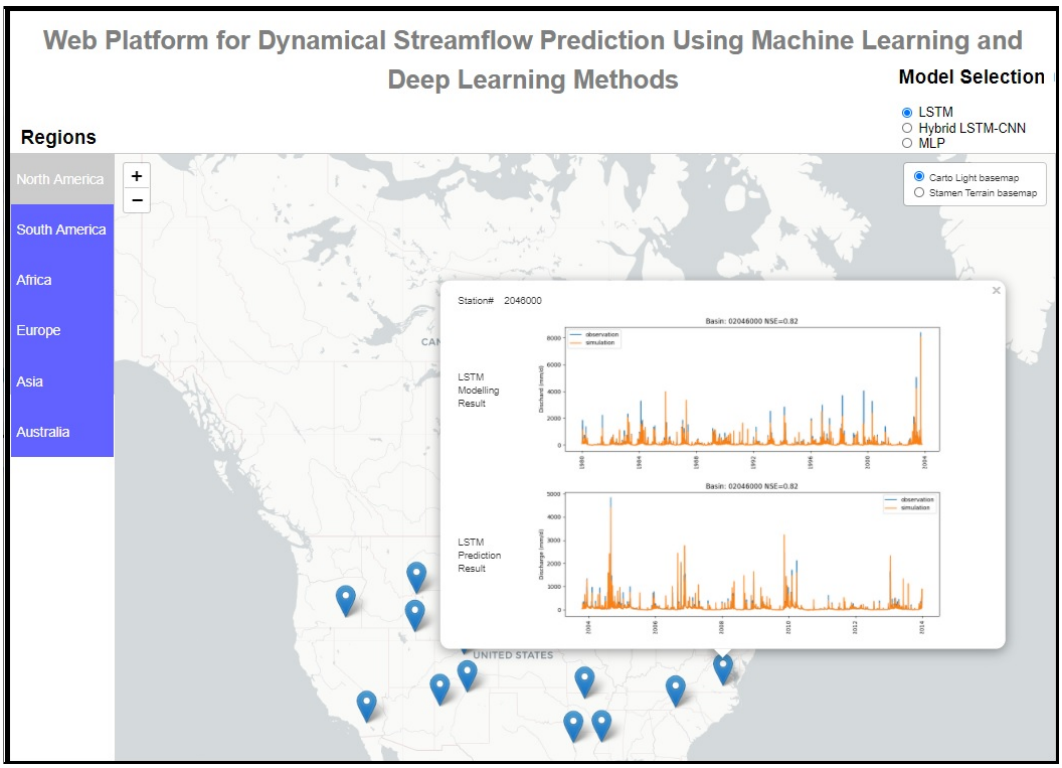


Figure 1: The new proposed web platform, North America case study as an example

### 2.4 Performance Assessment Criteria

Because no one evaluation metric can fully capture the consistency, reliability, accuracy, and precision of a streamflow model, it was necessary to use a variety of performance metrics for model benchmarking (Hoshin Vijai Gupta et al., 1998). The metrics for model evaluation are the Nash–Sutcliffe efficiency (NSE; Nash and Sutcliffe, 1970), the three decompositions following Hoshin V Gupta et al. (2009) which are the correlation coefficient of the observed and simulated discharge (r), the variance bias (α) and the total volume bias (β). These three metrics are combined and presented in the Kling-Gupta efficiency (KGE; Gupta et al., 2009) metric presented in equation 2. And the third criterion to evaluate the errors in simulations is root mean squared error (TRMSE) presented in equation 3. in these equation $Q_s$ is the simulated runoff, $Q_o$ observed runoff.

$$NSE = 1 - \frac{\sum_{t=1}^{t}(Q_s^t - Q_o^t)^2}{\sum_{t=1}^{t}(Q_o^t - \bar{Q}_o)^2} \tag{1}$$

$$KGE = 1 - \sqrt{(cc-1)^2 + (\alpha-1)^2 + (\beta-1)^2} \tag{2}$$

$$TRMSE = \sqrt{\sum_{t=1}^{n}\frac{(Q_s^t - Q_o^t)^2}{n}}, \quad where: \quad Q = \frac{(1+Q)^\lambda - 1}{\lambda} \tag{3}$$

## 3 Experimental Results

### 3.1 Preprocessing

All three models that were mentioned in the previous section were trained and tested using available data sets for the sample watersheds (18 watersheds from North America, 8 watersheds from South America and 6 watersheds from Africa). In order to find the optimal selection of stations,

geospatial analysis on station metadata was applied. Among coastal stations and basins with rich rainfall,streamflow data, we localized stations using unsupervised clustering approach such as k-means and self organizing map. NCDC and GRDC stations that were geographically closer to each other were paired up together to get the final input feature set. Each watershed dataset was split into 70:30 ratio for training and testing samples, respectively. Primary input features for all three models were daily precipitation, streamflow value at previous time step and current time step. Those input features were processed by replacing missing values and normalized with min-max normalization.

### 3.2 Model Setup and Hyperparameter Configuration

All three models were trained and tested with uniform set of hyperparameters and experimental run configuration. In order to find the optimal set of hyperparameters, an exhaustive search over specified parameter values for our regressor models were executed. GridSearch class from scikit-learn libraries was used for our purpose. The parameters for model selection were batch size, training epochs, optimizer, learning rate, weight initialization, activation function for convolution and LSTM layers, and last but not the least dropout rates. It is worth mentioning that dropping recurrent connections rather than activations yielded better performance for LSTM networks. Best performance gains were achieved with 50 epochs, SGD and RMSProp optimizer, learning rate of 0.001, ReLU activation. The main problems of ML and DL regression-based models is that they are prone to overfitting to the training data and thus perform poorly when given unseen data. In order to overcome this problem we considered 0.2 for the dropout rate.

### 3.3 Model Assessment

Models were assessed with collection of 18 watersheds from North America, 8 watersheds from South America and 6 watersheds from Africa continent. Streamflow prediction results are presented in the figure 2 for one watershed per case study region as an example to show the performance of each model.

Also, the performance assessment metrics are calculated for the simulation results of the three sample watersheds for the test period (table 1). The results showed that there is no significant difference among the models' performance but based on the table 1 we can say that MLP model provided the weakest results. The reason for this could be the memory-less structure of MLP compared to LSTM and CNN-LSTM, so however we entered the antecedent day runoff as an input to play the role of memory for this algorithm, but the deep learning models (LSTM and CNN-LSTM) still have more reliable results as they are instance-based algorithms. Also, for North America sample case, since there is a reasonable abundance of data, the models were able to learn considerably well and perform relatively well when tested using various metrics presented in table 1. Also in order to investigate the overall performance of different models the mean values of NSE metric for all the stations per region is also calculated and presented in the table 1.

Table 1: The calculated performance criteria for the simulated streamflow of one watershed in each region as an example as well as the NSE values average for all stations per region

| Region | Model | **NSE** | **KGE** | **TRMSE** | **NSE (Mean)** |
|---|---|---|---|---|---|
| North America | **MLP** | 0.82 | 0.88 | 0.98 | 0.79 |
| | **LSTM** | 0.82 | 0.87 | 1.11 | 0.8 |
| | **CNN-LSTM** | 0.84 | 0.9 | 0.95 | 0.81 |
| South America | **MLP** | 0.85 | 0.9 | 0.67 | 0.6 |
| | **LSTM** | 0.86 | 0.9 | 0.66 | 0.6 |
| | **CNN-LSTM** | 0.86 | 0.9 | 0.66 | 0.62 |
| Africa | **MLP** | 0.75 | 0.79 | 0.79 | 0.75 |
| | **LSTM** | 0.74 | 0.82 | 0.81 | 0.77 |
| | **CNN-LSTM** | 0.7 | 0.69 | 1.74 | 0.77 |

In addition, in order to investigate the performance of all three models over globe for train and test period, the cumulative probability function (CDF) for all NSE values is plotted in the figure 03. As it is obvious in this figure, the calculated NSE values was more than 0.8 for more than 50 percent of the

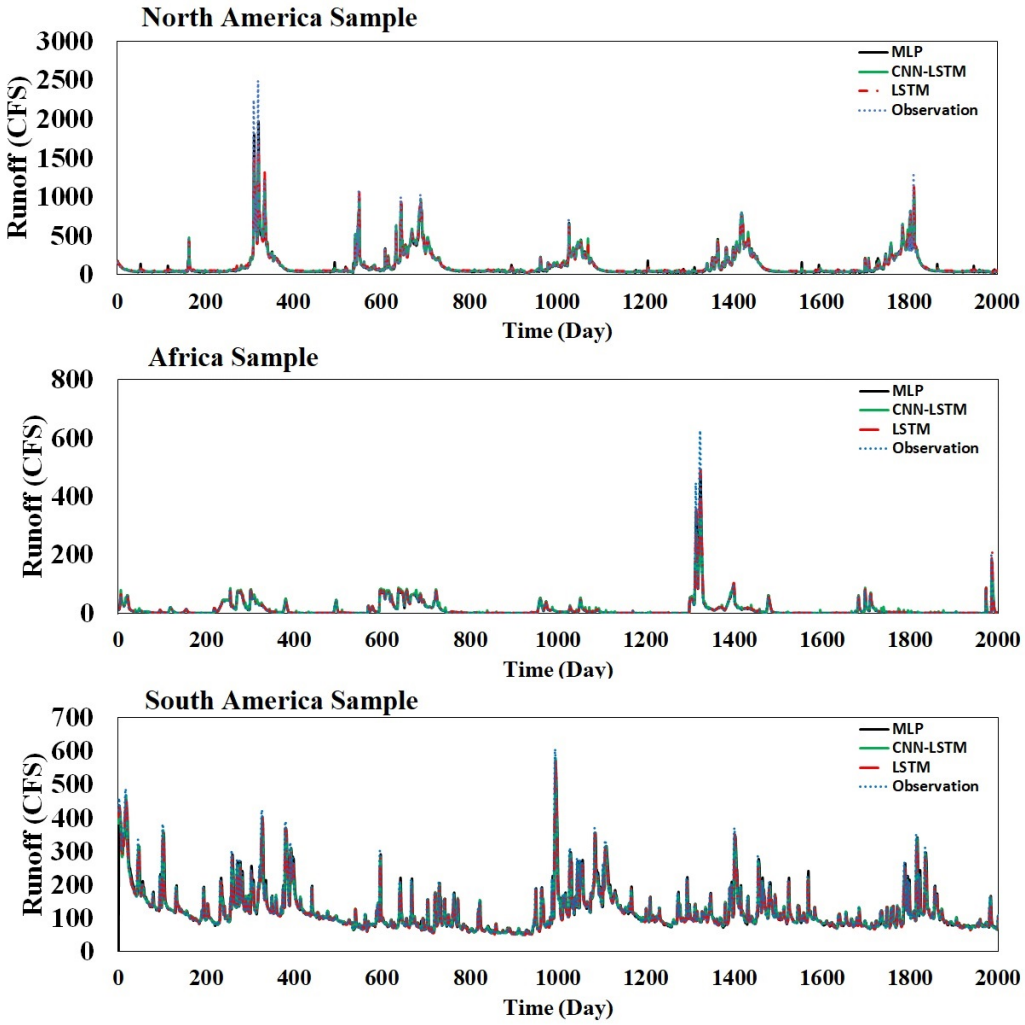


Figure 2: Streamflow prediction results for one watershed per case study region as an example for the test period

cases for both train and test periods which shows a successful performance of data driven algorithm in runoff modeling.

### 3.4 Future Forecasting

In order to forecast streamflow for future, we modeled encoder -decoder seq2seq LSTM network to automatically learn patterns from sequence time series and produce multi step forecasted time series for next 3 months. An encoder-decoder sequence-to-sequence LSTM is a model comprised of two sub networks: encoder changes input sequence to a fixed-length internal representation, and decoder uses it to predict the output sequence. We processed the input sequence by making stationary and transform to make it work as a supervised learning approach. We evaluated model using walk forward validation where daily streamflow data for few months were made available to make prediction on the subsequent months for future time steps. We demonstrated forecast results for 3 months on one station from North America region and it is included in Appendix 3 section.

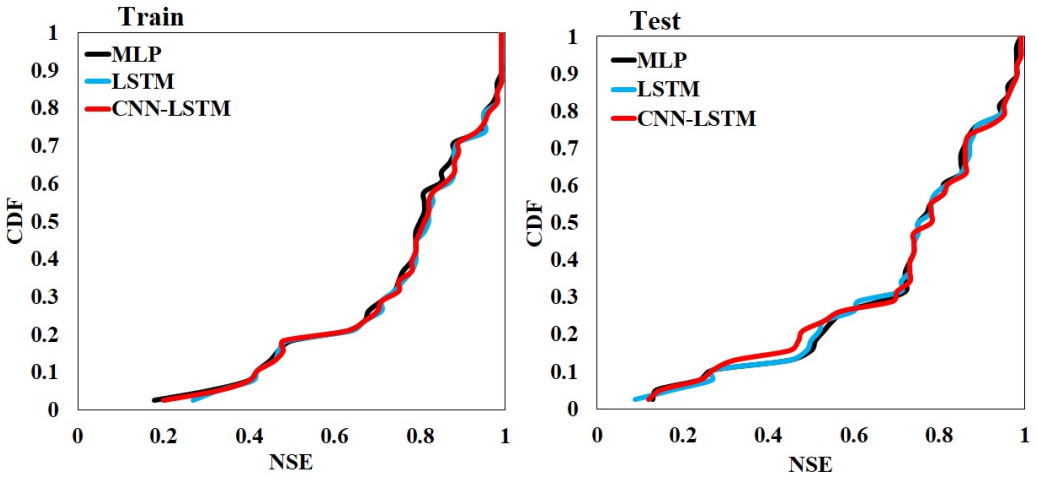


Figure 3: Cumulative Density Function (CDF) for the calculated NSE values over all test cases in train and test periods.

## 4 Conclusion

This study provided a web platform which employs simple to advanced regression based supervised learning for simulation rainfall-runoff processes over globe. Supervised machine learning means we have examples (rows) with input and output variables (columns). The algorithm uses function approximation to map inputs to outputs on specific prediction task in such a way that it has skill. The results showed daily streamflow simulation is improved with ML and DL regression supervised learning algorithms. In general, our analysis indicates that supervised learning can successfully simulate streamflow based on various metrics. The results showed that there is no significant difference among the models' performance but based on the table 1 we can say that MLP model provided the weakest results. The reason for this could be the memory-less structure of MLP compared to LSTM and CNN-LSTM, so however we entered the antecedent runoff as an input to play the role of memory for this algorithm, but the deep learning models (LSTM and CNN-LSTM) still have more reliable results as they are instance based algorithms. It is well-known that machine learning models could fail in simulating streamflows from only meteorological variables in the absence of antecedent streamflow values. The main reason for this could be low and lagged relationships between streamflow and meteorological variables (Tongal and Booij, 2018). To overcome this inefficiency, future works should focus on separating streamflow into different components such as base flow and surface flow to improve simulation and forecasting capabilities of machine learning models.

## References

[1] N. Addor, A. J. Newman, N. Mizukami, and M. P. Clark. The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences (HESS)*, 21(10):5293–5313, 2017.

[2] B. Asadieh and N. Y. Krakauer. Global change in streamflow extremes under climate change over the 21st century. *Hydrology and Earth System Sciences*, 21(11):5863, 2017.

[3] D. T. Bui, B. Pradhan, H. Nampak, Q.-T. Bui, Q.-A. Tran, and Q.-P. Nguyen. Hybrid artificial intelligence approach based on neural fuzzy inference model and metaheuristic optimization for flood susceptibilitgy modeling in a high-frequency tropical cyclone area using gis. *Journal of Hydrology*, 540:317–330, 2016.

[4] J. L. Campbell, C. T. Driscoll, A. Pourmokhtarian, and K. Hayhoe. Streamflow responses to past and projected future changes in climate at the hubbard brook experimental forest, new hampshire, united states. *Water Resources Research*, 47(2), 2011.

[5] K. Fang, C. Shen, D. Kifer, and X. Yang. Prolongation of smap to spatiotemporally seamless coverage of continental us using a deep learning neural network. *Geophysical Research Letters*, 44(21):11–030, 2017.

[6] H. V. Gupta, S. Sorooshian, and P. O. Yapo. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751–763, 1998.

[7] H. G. Hidalgo, T. Das, M. D. Dettinger, D. R. Cayan, D. W. Pierce, T. P. Barnett, G. Bala, A. Mirin, A. W. Wood, C. Bonfils, et al. Detection and attribution of streamflow timing changes to climate change in the western united states. *Journal of Climate*, 22(13):3838–3855, 2009.

[8] K. Kasiviswanathan, J. He, K. Sudheer, and J.-H. Tay. Potential application of wavelet neural network ensemble to forecast streamflow for flood management. *Journal of Hydrology*, 536:161–173, 2016.

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

[10] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[11] Z. Liu, W. Xu, J. Feng, S. Palaiahnakote, T. Lu, et al. Context-aware attention lstm network for flood prediction. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1301–1306. IEEE, 2018.

[12] E. P. Maurer, A. W. Wood, J. C. Adam, D. P. Lettenmaier, and B. Nijssen. A long-term hydrologically based dataset of land surface fluxes and states for the conterminous united states. *Journal of climate*, 15(22):3237–3251, 2002.

[13] J. E. Nash and J. V. Sutcliffe. River flow forecasting through conceptual models part i—a discussion of principles. *Journal of hydrology*, 10(3):282–290, 1970.

[14] A. Newman, M. Clark, K. Sampson, A. Wood, L. Hay, A. Bock, R. Viger, D. Blodgett, L. Brekke, J. Arnold, et al. Development of a large-sample watershed-scale hydrometeorological data set for the contiguous usa: data set characteristics and assessment of regional variability in hydrologic model performance. *Hydrology and Earth System Sciences*, 19(1):209, 2015.

[15] S. Sadeghi-Tabas, S. Samadi, B. Zahabiyoun, et al. Application of bayesian algorithm in continuous streamflow modeling of a mountain watershed. *European Water*, 57:101–108, 2017.

[16] J. Sadler, J. Goodall, M. Morsy, and K. Spencer. Modeling urban coastal flood severity from crowd-sourced flood reports using poisson regression and random forest. *Journal of hydrology*, 559:43–55, 2018.

[17] S. Samadi and M. Meadows. The transferability of terrestrial water balance components under uncertainty and nonstationarity: A case study of the coastal plain watershed in the southeastern usa. *River Research and Applications*, 33(5):796–808, 2017.

[18] S. Samadi, M. Pourreza-Bilondi, C. Wilson, and D. Hitchcock. Bayesian model averaging with fixed and flexible priors: Theory, concepts, and calibration experiments for rainfall-runoff modeling. *Journal of Advances in Modeling Earth Systems*, 12(7):e2019MS001924, 2020.

[19] S. Samadi, D. Tufford, and G. Carbone. Assessing parameter uncertainty of a semi-distributed hydrology model for a shallow aquifer dominated environmental system. *JAWRA Journal of the American Water Resources Association*, 53(6):1368–1389, 2017.

[20] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[21] C. Shen. A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, 54(11):8558–8593, 2018.

[22] C. Shen, E. Laloy, A. Elshorbagy, A. Albert, J. Bales, F.-J. Chang, S. Ganguly, K.-L. Hsu, D. Kifer, Z. Fang, et al. Hess opinions: Incubating deep-learning-powered hydrologic science advances as a community. *Hydrology and Earth System Sciences (Online)*, 22(11), 2018.

[23] C. Sivapragasam and N. Muttil. Discharge rating curve extension–a new approach. *Water Resources Management*, 19(5):505–520, 2005.

[24] P. E. Thornton, M. M. Thornton, B. W. Mayer, N. Wilhelmi, Y. Wei, R. Devarakonda, and R. Cook. Daymet: Daily surface weather on a 1 km grid for north america, 1980-2008. *ddsw*, 2012.

[25] H. Tongal and M. J. Booij. Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. *Journal of hydrology*, 564:266–282, 2018.

[26] Y. Xia, K. Mitchell, M. Ek, B. Cosgrove, J. Sheffield, L. Luo, C. Alonge, H. Wei, J. Meng, B. Livneh, et al. Continental-scale water and energy flux analysis and validation for north american land data assimilation system project phase 2 (nldas-2): 2. validation of model-simulated streamflow. *Journal of Geophysical Research: Atmospheres*, 117(D3), 2012.

[27] S. Yang, D. Yang, J. Chen, J. Santisirisomboon, W. Lu, and B. Zhao. A physical process and machine learning combined hydrological model for daily streamflow simulations of large watersheds with limited observation data. *Journal of Hydrology*, 590:125206, 2020.

## Appendix 1: LSTM Model in-detail Description

**Long Short-Term Memory (LSTM)**

The LSTM block consists of an input gate, a memory cell, a forget gate, and an output gate. First, LSTM decides what data should be removed from the cell state. This decision is made by a sigmoid layer called the forget gate, the equation for which is show below:

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + b_f) \tag{4}$$

The previous moment output $h_{t-1}$ and current input $X_t$ enable the forget gate to generate an $f_t$ value between 0 and 1. This decides whether the information produced in the previous moment $C_t$ passes or partially passes.

Next, it decides what new data will be saved in the cell state, which is done in two steps: First, the input gate determines which values will be updated. Second, the memory cell creates a vector of new candidate values $C_t$, which can be added to the state. The values produced by these two parts will later be combined to update the input gate, seen in equation 5, as well as the cell state, seen in equation 6

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i) \tag{5}$$

$$C_t = f_t \times C_{t-1} + i_t \times tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

Finally, the output of the model is determined. An initial output is obtained through the sigmoid layer (output gate), and then the $C_t$ value is resized to be between -1 and 1 through $tanh$ and multiply it by the output of the sigmoid gate so that we only output the target parts, which are the output gate, seen in equation 7, and the hidden state, seen in equation 8

$$O_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o) \tag{7}$$

$$h_t = O_t \times tanh(C_t) \tag{8}$$

In Equations 4- 7, $\sigma$ and $tanh$ represent the gate activation function and the hyperbolic tangent activation function, respectively. $W$ represents the corresponding weight coefficient matrix.

## Appendix 2: Complementary Results

### North America

### MLP

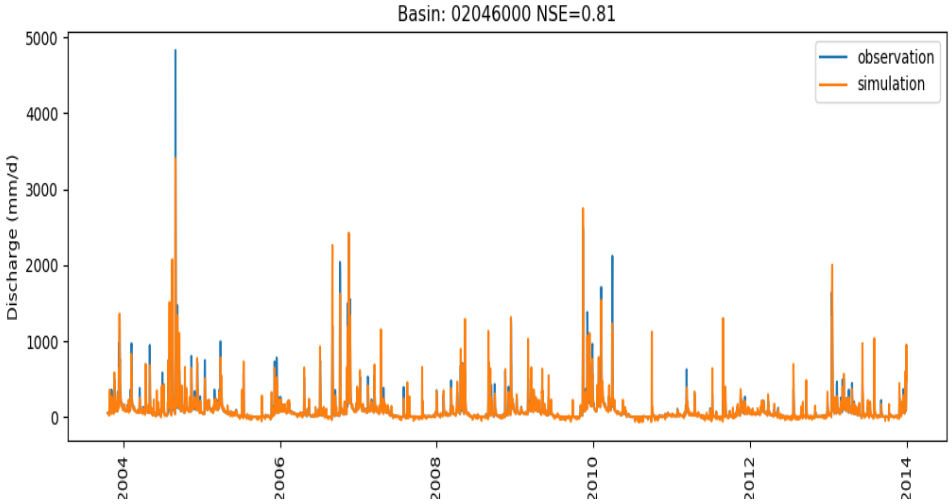


Figure 4: MLP simulation as well as observation for the test period

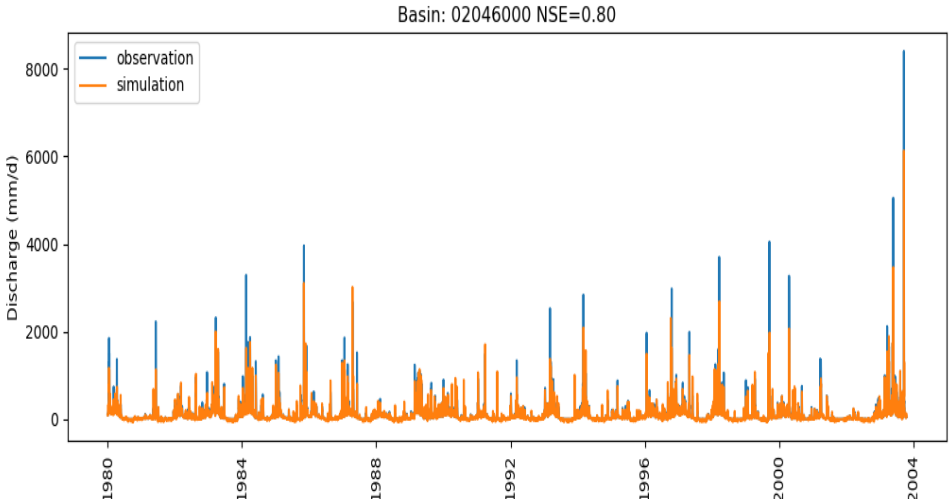


Figure 5: MLP simulation as well as observation for the train period

**LSTM**

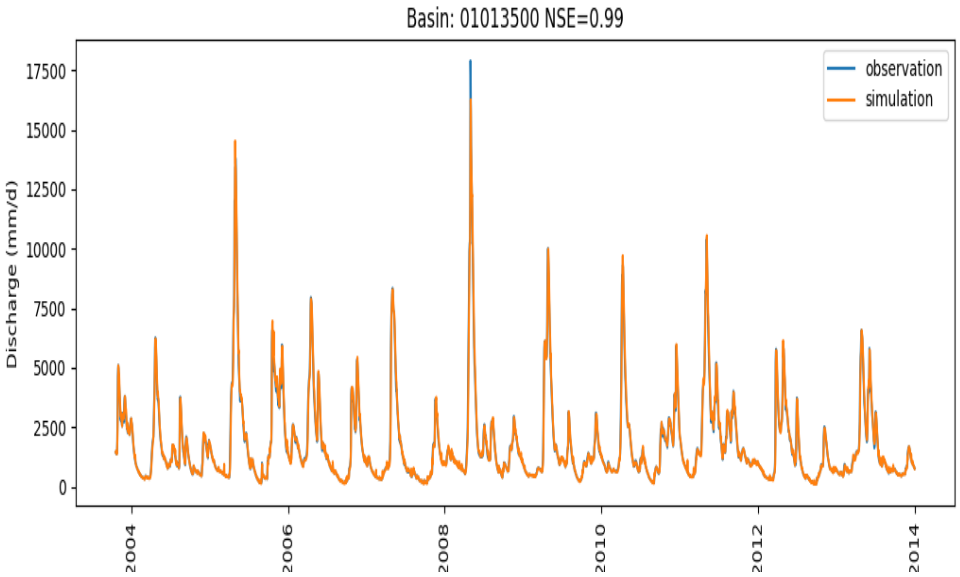


Figure 6: LSTM simulation as well as observation for the test period

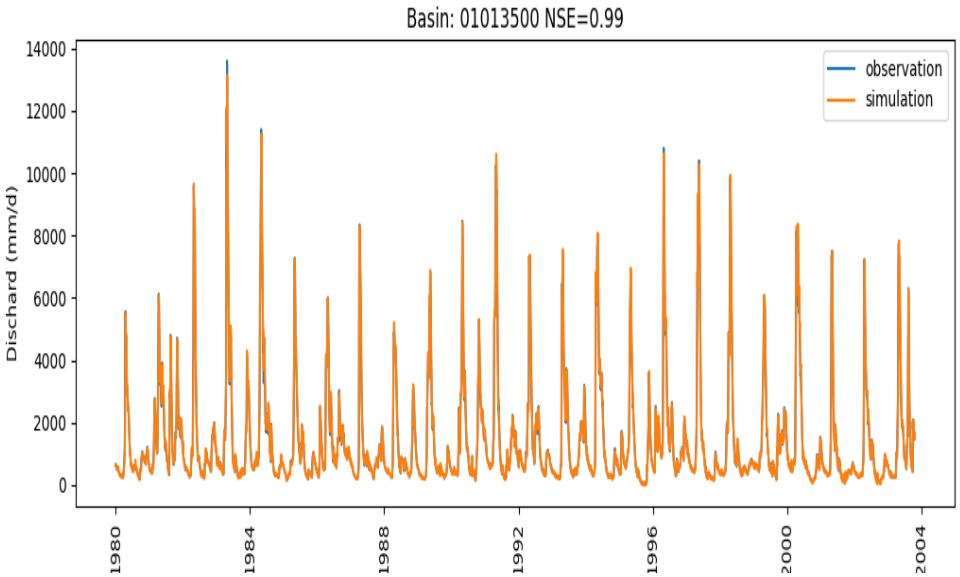


Figure 7: LSTM simulation as well as observation for the train period

**CNN-LSTM**

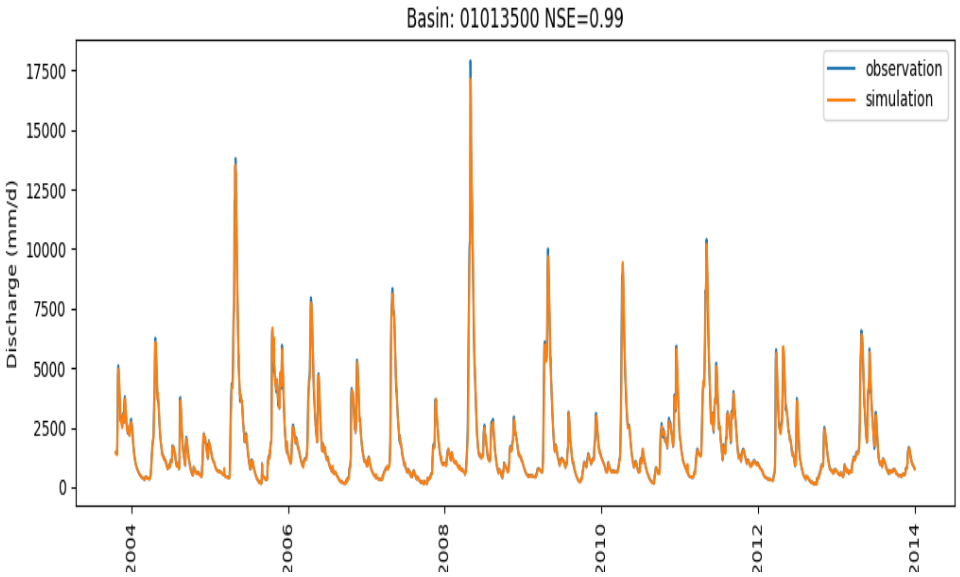


Figure 8: CNN-LSTM simulation as well as observation for the test period

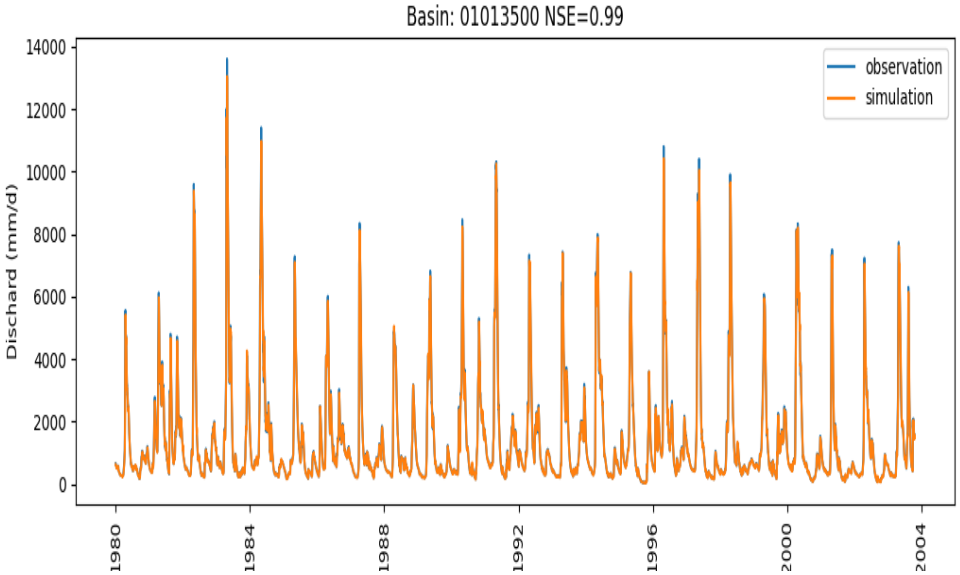


Figure 9: CNN-LSTM simulation as well as observation for the train period

**South America**

**MLP**

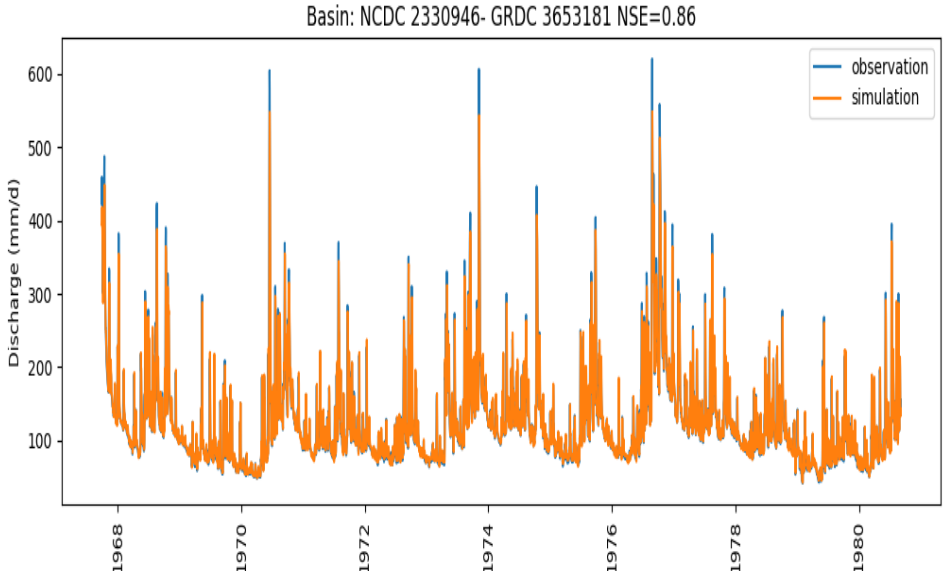


Figure 10: MLP simulation as well as observation for the test period

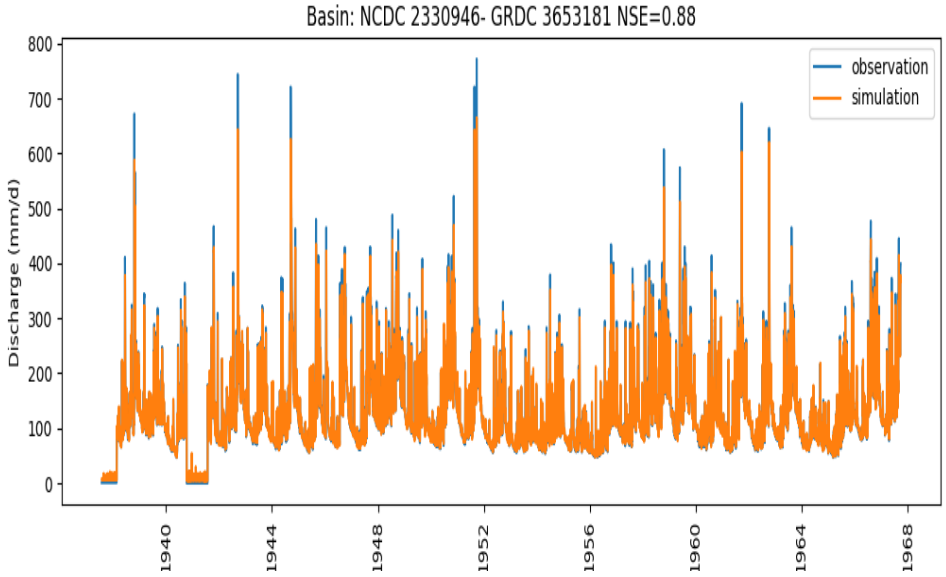


Figure 11: MLP simulation as well as observation for the train period

**LSTM**

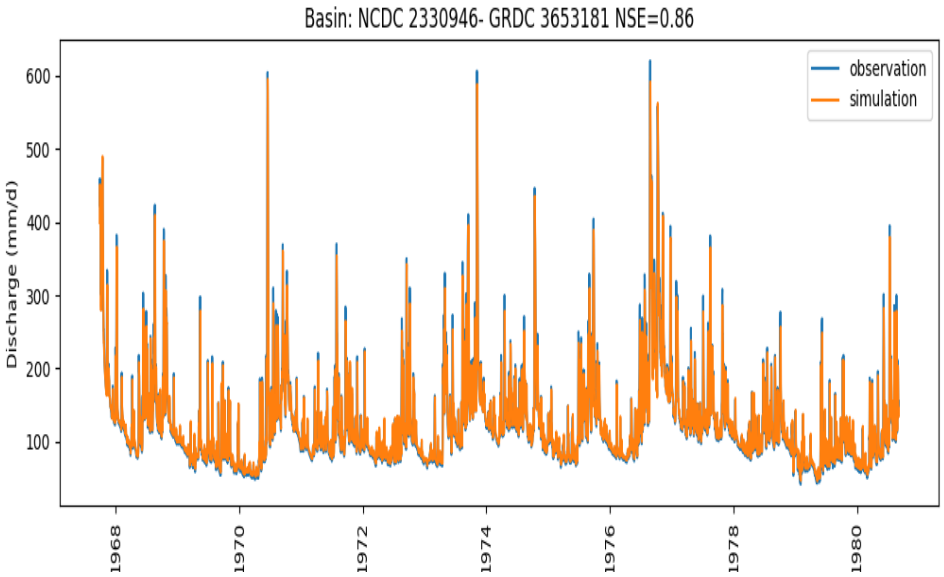


Figure 12: LSTM simulation as well as observation for the test period

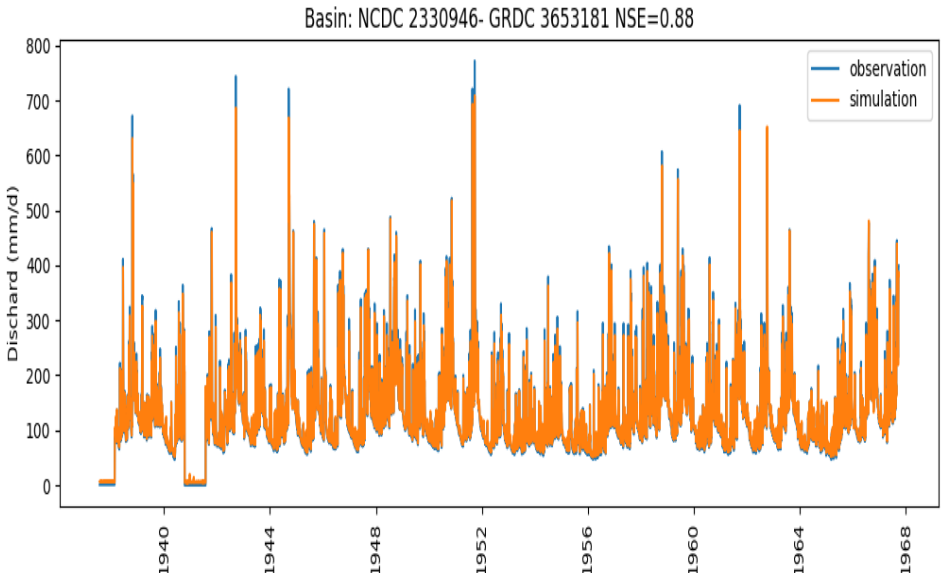


Figure 13: LSTM simulation as well as observation for the train period

**CNN-LSTM**

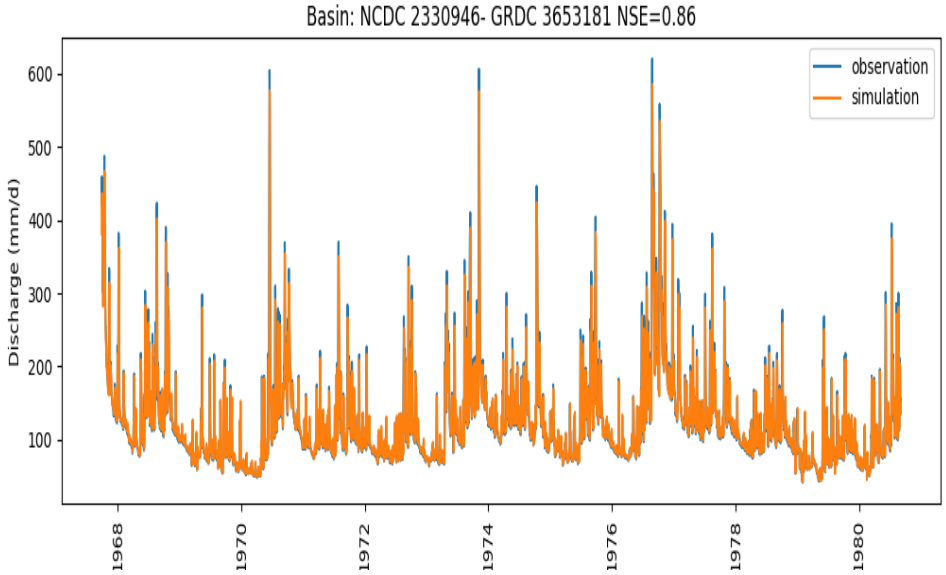


Figure 14: CNN-LSTM simulation as well as observation for the test period

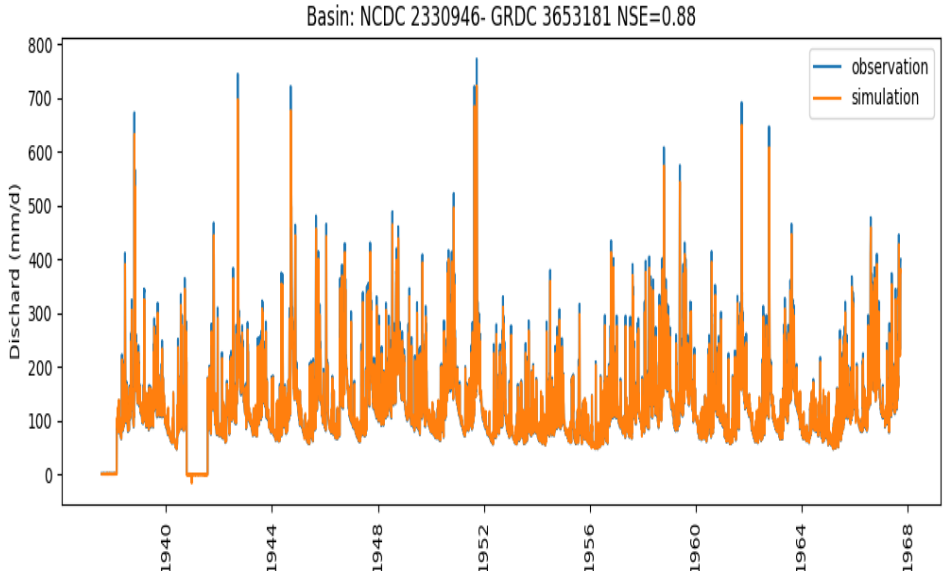


Figure 15: CNN-LSTM simulation as well as observation for the train period

**Africa**

**MLP**

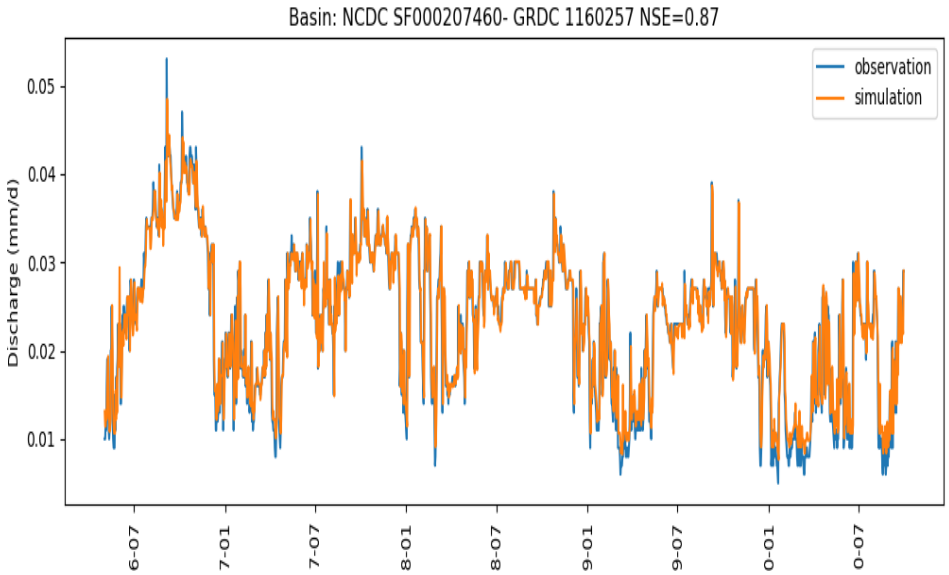


Figure 16: MLP simulation as well as observation for the test period

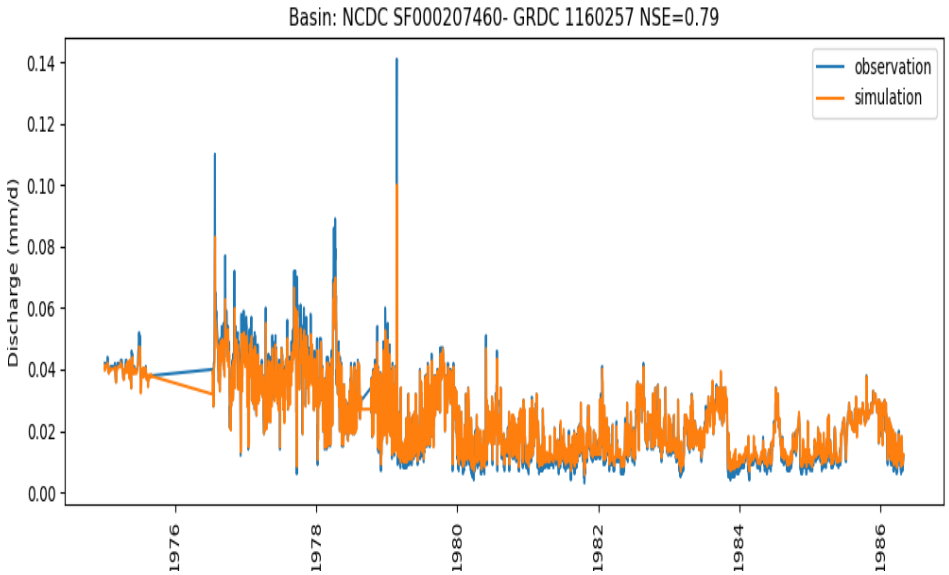


Figure 17: MLP simulation as well as observation for the train period

**LSTM**

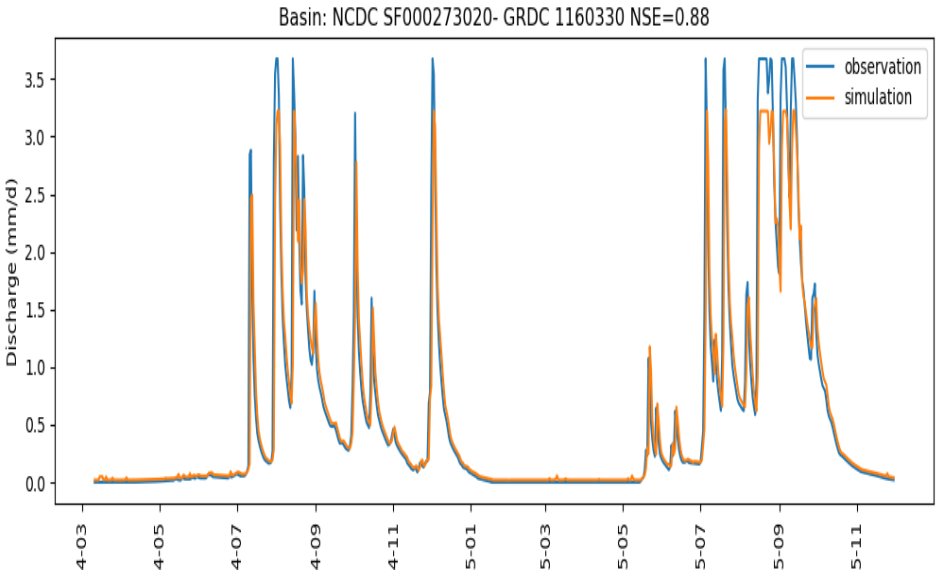


Figure 18: LSTM simulation as well as observation for the test period

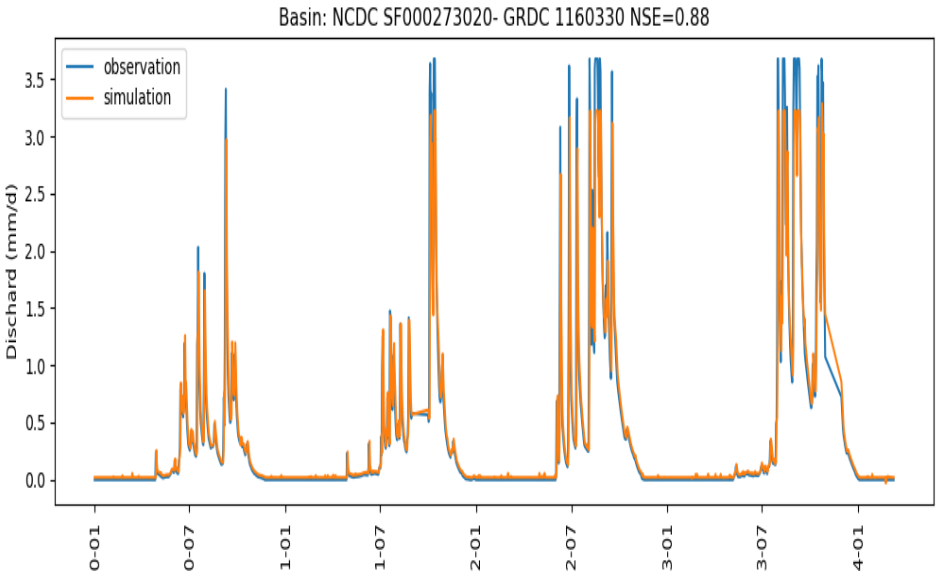


Figure 19: LSTM simulation as well as observation for the train period

**CNN-LSTM**

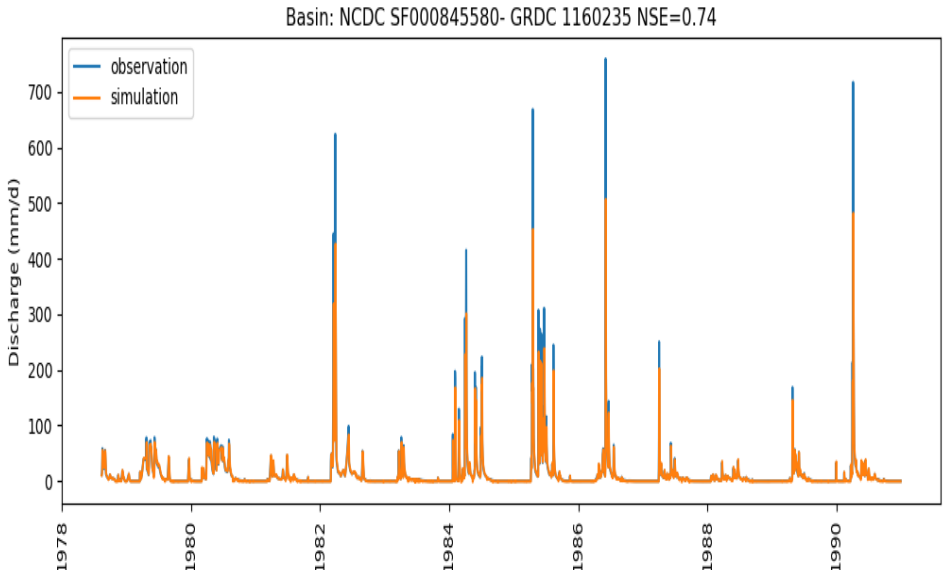


Figure 20: CNN-LSTM simulation as well as observation for the test period

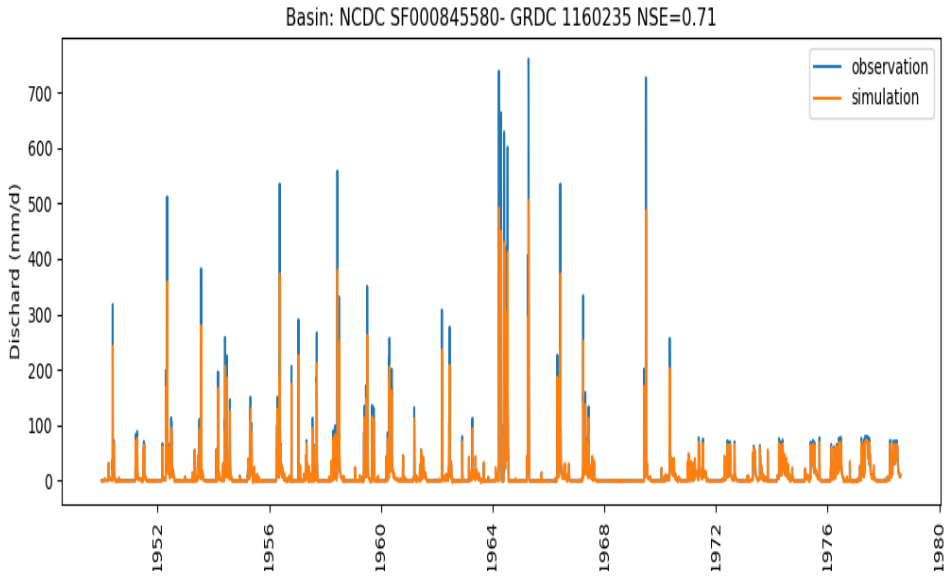


Figure 21: CNN-LSTM simulation as well as observation for the train period

## Appendix 3: Forecast results

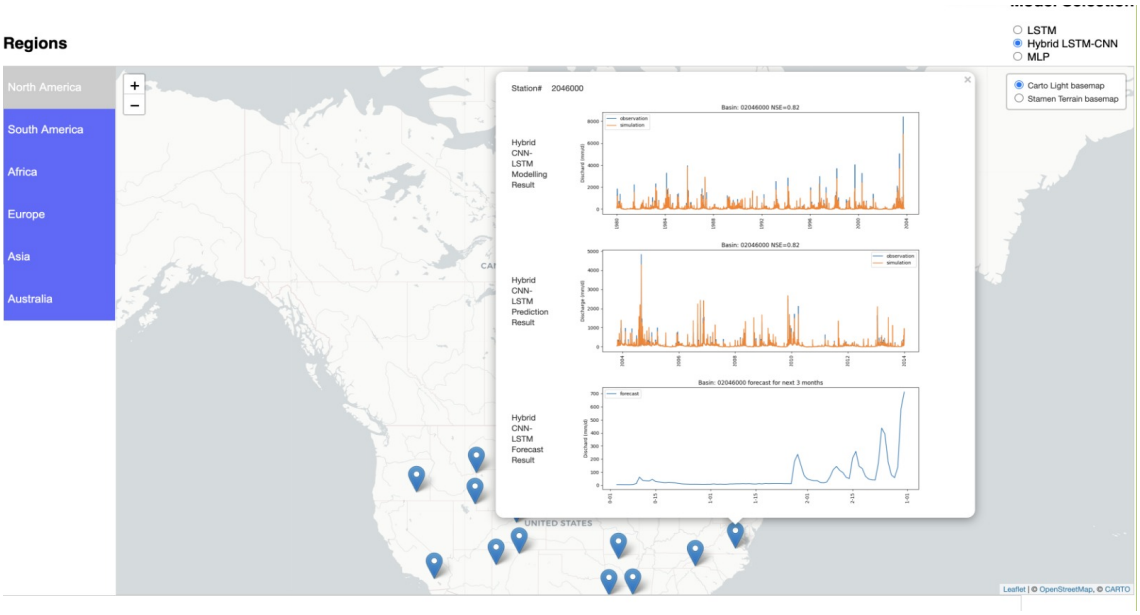


Figure 22: Next 3 months streamflow forecast for Basin 0204600