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@clemson.edu

Siddish P Rao
siddisr@g.clemson.edu

Meghan Patil
mmpatil@clemson.edu

October 31, 2020

1 Introduction

1.1 Problem Definition

Watershed simulations help understand the past and current state of rainfall-runoff processes in the basin 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 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 widely applied for streamflow simulations and forecasting.

1.2 Literature Review

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 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 flow 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 and South America and Africa) using various state-of-the-art ML and deep learning (DL) models as data-driven approaches. Data-driven approach 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 combination of LSTM and Convelutional Neural Network (CNN). This research is going to provide a web platform modeling framework in global scale using the google cloud technology. In order to overcome the curse of modeling and massive data processing we are going to take advantage of GPU-Accelerated Google Cloud Platform. To do so, first the preprocessing of the avalable datasets has been done in order to impute the missing values with an accurate estimation using diiferent methods (please see the report for the checkpoint one). Then the mentioned data drivon methods and updated datasets has deployed on google platform (we are still working on this step). Finally, the implemented models, trained and tested for all case studies. The detailed description of process and the challenges in this research project is presented in the following sections including metodology and results.

2 Methodology

2.1 Case studies

2.1.1 United States of America

The underlying data for USA 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 US catchment averages and in order to have similar 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 US 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 imputed and data 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 imputed and data used to train and test the models employed in this study.

2.2 Web Application

The main web applcation was done in Python with the help of ArcGIS API for python. This API provides interactive display of geographic information through a composition of web layers, basemap and other similar tools for geographic mapping. We also used JavaScript to implement certain UI elements like popups and legend for the layers in the application.

The HUC-4 watershed shapefiles for entire case studies uploaded in the ArcGIS online website as a public layer. Also, the rainfall and runoff stations and its associated dataset uploded in ArcGIS online website as feature layers and then all information deployed in the web app using APIs.

A basic preview of the proposed web app is presented in the Figures 1-3 for USA, South America and Africa respectively.

We are still working on this part of the project, new update coming soon ...

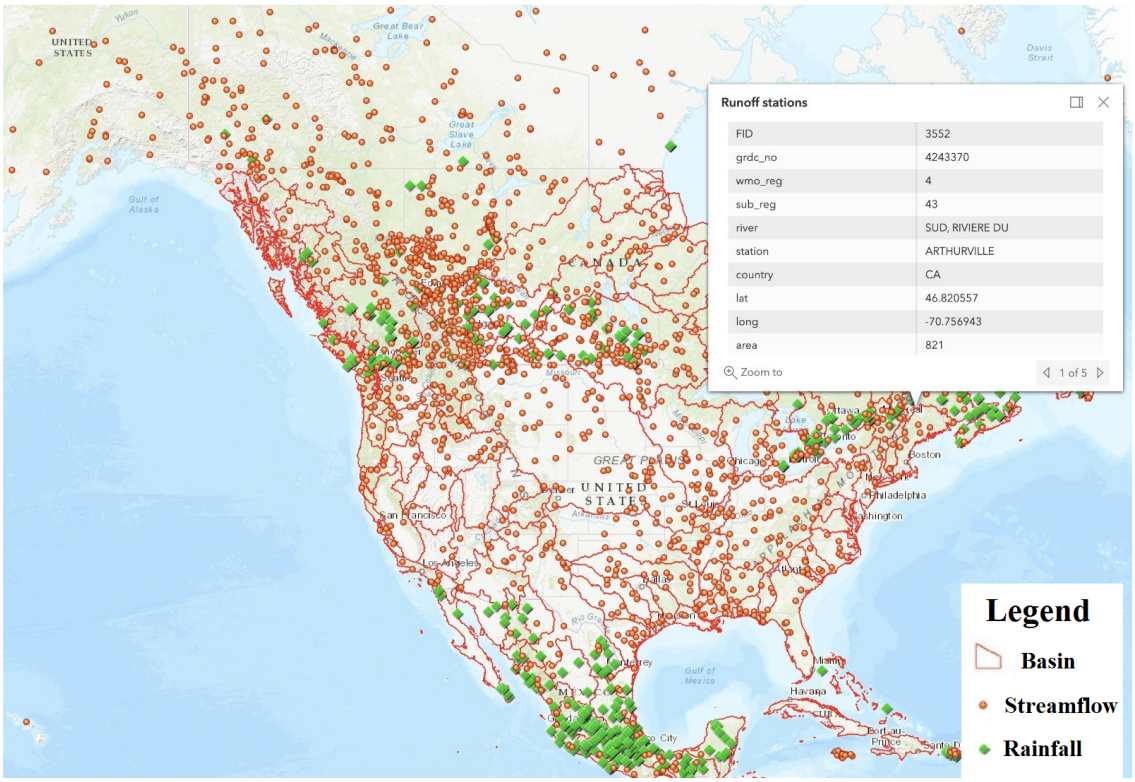

Figure 1: The new proposed web platform, USA case study

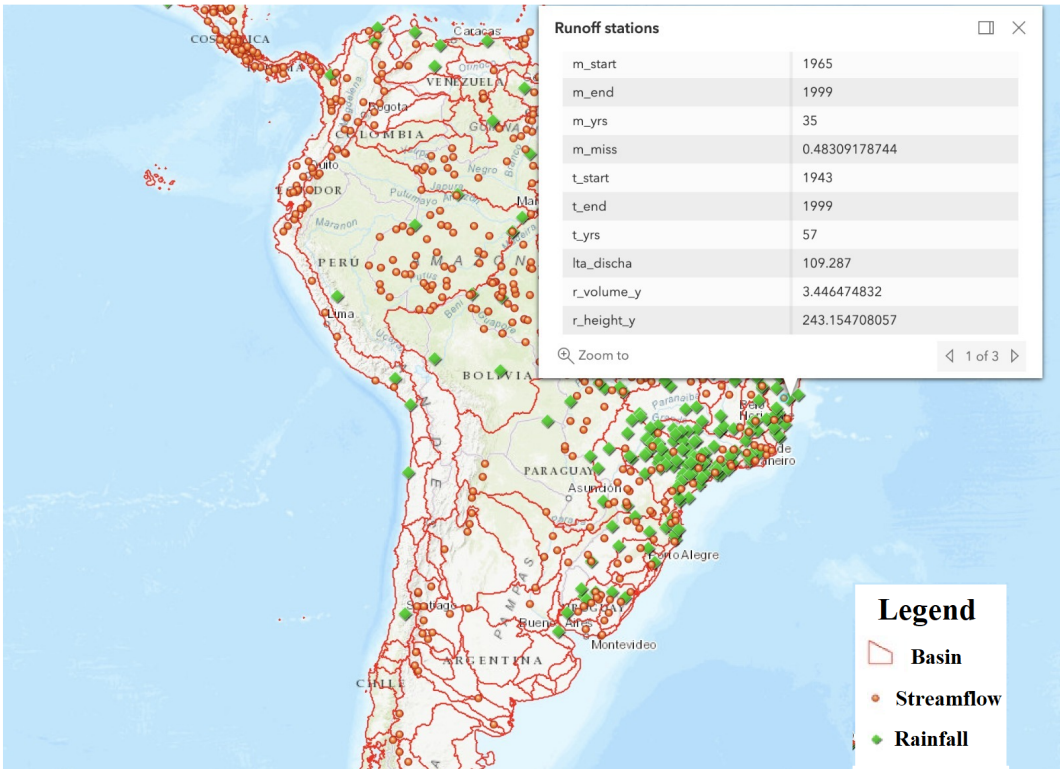

Figure 2: The new proposed web platform, North America case study

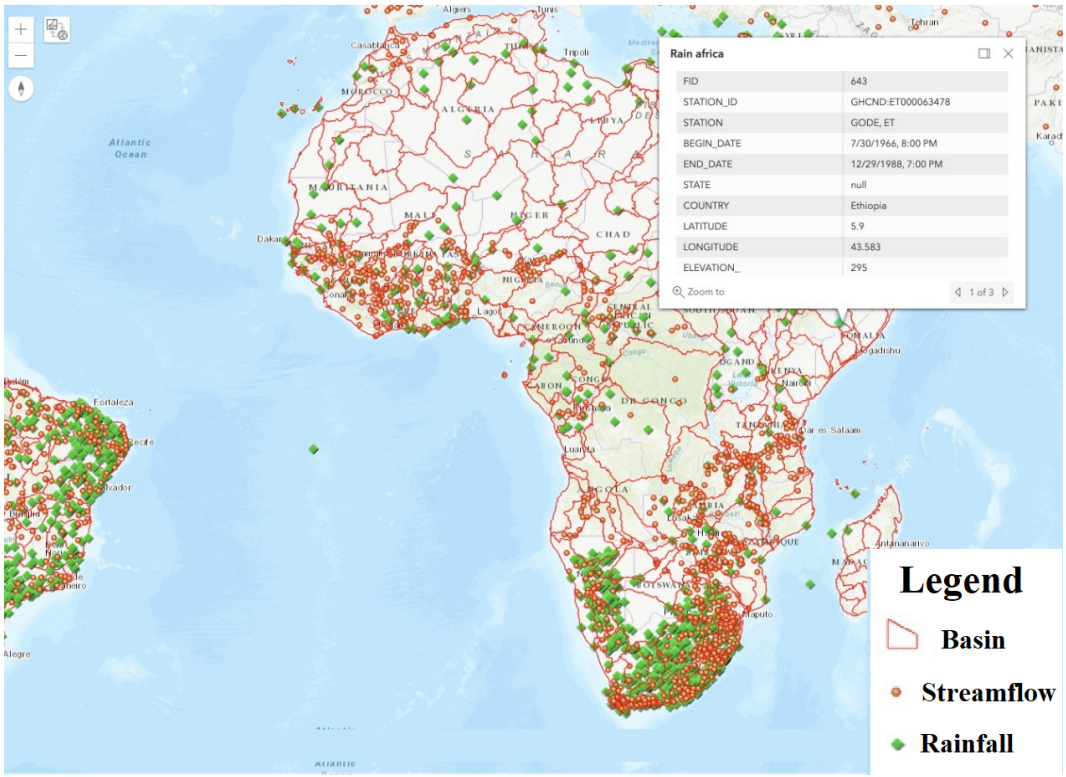

Figure 3: The new proposed web platform, Africa case study

2.3 Simulation Models

2.3.1 Multilayer 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.

2.3.2 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.

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:

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

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 2, as well as the cell state, seen in equation 3

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i) \tag{2}$$

$$C_t = f_t \times C_{t-1} + i_t \times tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

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 4, and the hidden state, seen in equation 5

$$O_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o) \tag{4}$$

$$h_t = O_t \times tanh(C_t) \tag{5}$$

In Equations 1- 4, σ and $tanh$ represent the gate activation function and the hyperbolic tangent activation function, respectively. W represents the corresponding weight coefficient matrix.

2.3.3 Convolutional Neural Network-LSTM (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.

2.4 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 7. And the third critrion to evaluate the erros in simulations is root mean squared error (RMSE) presented in equation 8. in these equation Q_s is the simulated runoff, Q_o observed runoff.

$$NSE = 1 - \frac{\sum_{t=1}^{t}(Q_s^t - Q_o^t)^2}{\sum_{t=1}^{t}(Q_o^t - \bar{Q}_o)^2} \tag{6}$$

$$KGE = 1 - \sqrt{(cc-1)^2 + (\alpha-1)^2 + (\beta-1)^2} \tag{7}$$

$$RMSE = \sqrt{\sum_{t=1}^{n} \frac{(Q_s^t - Q_o^t)^2}{n}} \tag{8}$$

3 Result and discussion

The mentioned models constructed as we discussed in the previous section and then trained and tested using available data sets for the sample watersheds (18 watersheds from USA, 8 watersheds from South America and 6 watersheds from Africa). Figures 1-3 shows the streamflow simulated by each method for the test period for USA, South America and Africa, respectively (we just briefly presented the prediction results for one watershed per case study as an example).

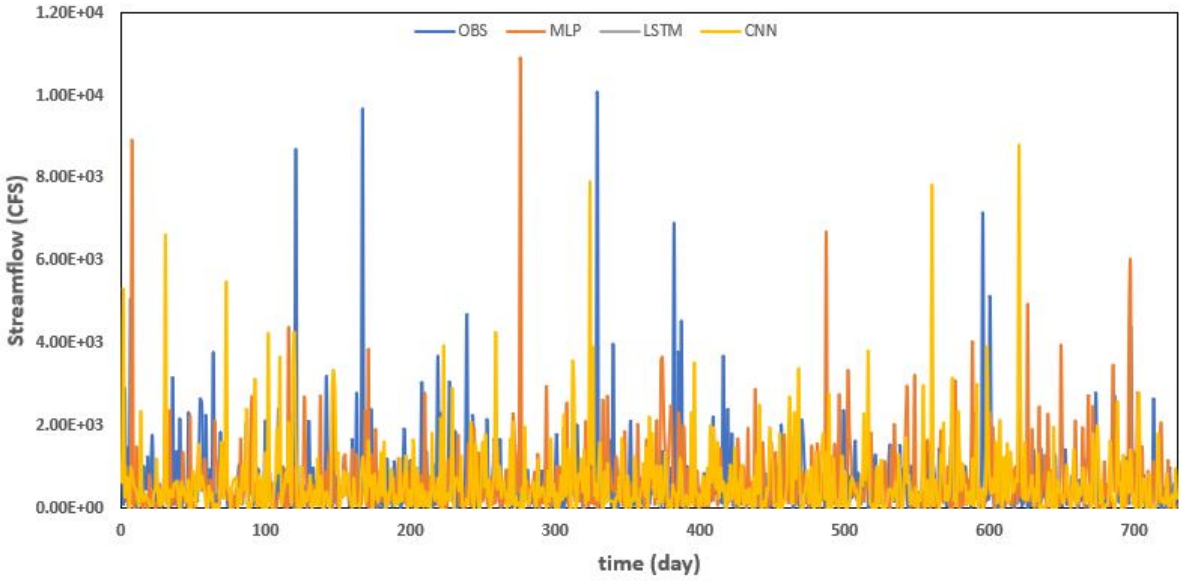

Figure 4: Streamflow simulation using MLP, LSTM and CNN-LSTM, USA case study

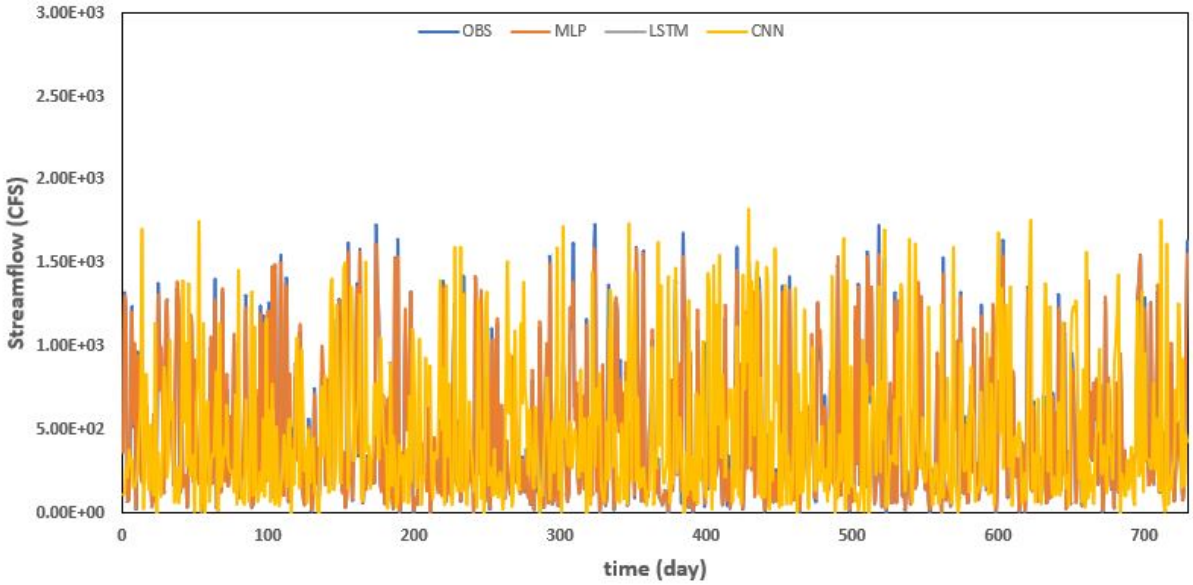

Figure 5: Streamflow simulation using MLP, LSTM and CNN-LSTM, South America case study

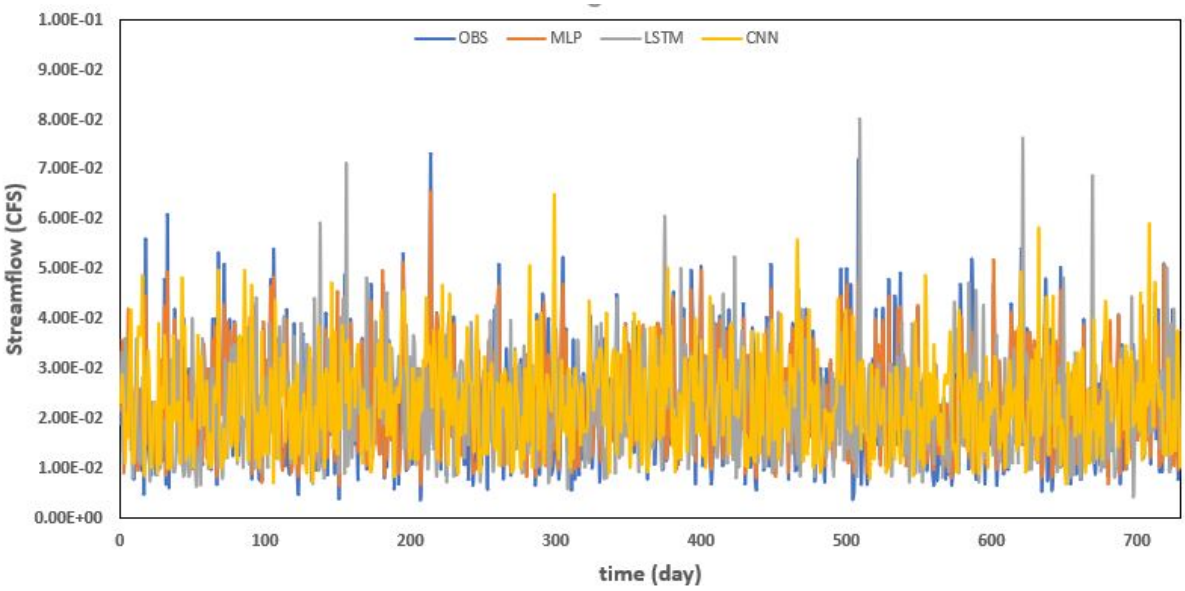

Figure 6: Streamflow simulation using MLP, LSTM and CNN-LSTM, Africa case study

Also, the performance criteria for the three selected watersheds (in the test period) are calculated and presented in the table 1.

Region	Model	**NSE**	**KGE**	**RMSE**
USA	**MLP**	0.68	0.71	70.5
	LSTM	0.78	0.8	62.68
	Conv-LSTM	0.79	0.8	61.32
South America	**MLP**	0.67	0.69	61.6
	LSTM	0.72	0.74	48.2
	Conv-LSTM	0.72	0.74	49.1
Africa	**MLP**	0.65	0.67	0.02
	LSTM	0.71	0.72	0.02
	Conv-LSTM	0.71	0.73	0.02

Table 1: The calculated performance criteria for the simulated streamflow of one watershed in each region as an example

In order to analyse and show the performance of different methods in all selected watersheds from different region, we have calculated the cumulative density function (CDF) over NSE performance criteria computed in this study (Figures XX-XX).

We are still working on this part of the project, new update coming soon ...

4 Future Work

in the final report we are going to complete the web platform section and the last part of the results and discussion section, so as the last step we want to deploy everything on google cloud and use GPU-Accelerated Google Engine for our developed web application. Also, we are looking for some real time datasets in order to simulate and forecast streamflow in real-time.

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] 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.

[26] 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.