Dynamical Streamflow Prediction Using Machine Learning Methods

Sadegh Sadeghi Tabas
sadeghs@clemson.edu

Nushrat Humaira
nhumair@g.clemson.edu

Siddish P Rao
siddisr@g.clemson.edu

Pawan Madanan
pmadana@g.clemson.edu

Meghan Patil
mmpatil@g.clemson.edu

September 30th, 2020

1 Formal Problem Statement

Prediction of the streamflow at river basin is one of the key challenges in the field of hydrology. It has a long history in hydrological sciences and the first attempts to predict the discharge as a function of precipitation events using regression-type approaches date back 170 years [1, 6]. Accurate prediction of runoff is critical for reducing the risk of flooding and improve preparedness and planning emergency situations. With accurate modelling of catchment dynamics and streamflow, it could not only provide a flood warning to reduce hazards but also enhance proper reservoirs management during the drought periods. However, accurate forecasting of runoff is challenging due to the complexity of rainfall runoff processes and the drainage system. More specifically, rainfall-runoff process is non-linear and influenced by many factors such as river basin surface mantle, temporal and spatial variability of basin characteristics and rainfall, elevation and the catchment geographic setting. Many forecasting models require an enormous amount of data and simulation runs and time. Therefore, rainfall-runoff modelling is an old but still mostly outstanding problem in the hydrological research.

Currently, there are three approaches for runoff prediction: physical-based (white-box) models, conceptual (gray-box) models and data driven (black-box) models.Among these three types of models, the conceptual and physical-based are maybe the best two models to understand the process of rainfall-runoff. While these models generally tend to require a large amount of input data (i.e., rainfall, temperature and evapotranspiration data) which may not always be available or could be difficult to obtain. Therefore, data driven models have been increasingly emphasized during these years again.

These black box models are used more and more as the data-driven techniques are developing [7]. The Artificial Neural Networks (ANN), one of the data-driven techniques, have been widely used in hydrology as an alternative to physical-based and conceptual models [8, 9]. These ANN techniques are based on artificial intelligence (AI), which is among the most famous skills in recent years. These skills could capture non-linearity and non-stationarity related to hydrological applications. Thus, data-driven methods based on AI have gained more attention for rainfall-runoff simulation [2].

In the last two decades, AI has been widely used for efficient simulating of nonlinear systems and capturing noise complexity in the datasets. Comparing with the classical black box models such as Auto Regressive (AR), Moving Average (MA), Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA), Auto Regressive Integrated Moving Average with exogenous input (ARIMAX), Linear Regression (LR), and Multiple Linear Regression (MLR) which are linear, AI-based models are non-linear models which are able to capture non-stationarity and non-linearity features. As a result, more and more researchers have developed models that are able to overcome the drawbacks of conventional models [10].

All in all, based on what we explained above, this research project is drafted in two sections including: 1- Data preprocessing for the first checkpoint and 2- train a streamflow driven model for the checkpoint two. So as this report represents the first checkpoint of the project, in the following paragraphs, we are going to explain the data preprocessing section in detail and what was the challenges with filling the missing values in the dataset.

As the first step we have downloaded the streamflow data from the GRDC website. Our study regions are including South, North and Central America and Africa with more than 4,600 stations containing daily streamflow from around 1900 til today. The downloaded massive dataset contains missing values which need to be predicted (if it is less than 10 persent of whole available data for that specific station) or removed. To do so, we have implemented seven different methods including, AI Method, Long Short-Term Memory (LSTM) which is a new generation of machine learning methods (Deep Learning) developed by Hochreiter (1997) [4] and the Kalman filter combined with ARIMA method developed by Hyndman and Khandakar (2008) [5], filling using mean value, Moving Average method (MA), Interpolation and finally imputation by the observation. The proposed methodology and the results is presented in detail in the following sections.

2 Detailed Explanation of your Solution

2.1 Artificial Intelligence Method

AI and other deep learning tools are some of the most common methods used for variety of tasks, including filling missing values in dataset. We used the Keras library for Python language to implement an AI based method for filling missing values. The model used fills in the missing precipitation values for a particular target station by using the values from its neighbouring station to try and predict the values. The method involves using Keras Sequential model which is a linear stack of layers.

2.2 Long Short-Term Memory Network Method

To predict missing values in a time series, LSTM network works best. As each time step, daily or monthly we observe a change in the river runoff value which captures time based sequence and predicts runoff for future time step. For this checkpoint, we implemented a single LSTM layer network with 4 cells. We plan to improve this network with two time series as input, one starting from current time step and another one has one time step lagged.

2.3 Kalman Filter Combined with the ARIMA Method

Auto Regressive Integrated with Moving Average also called ARIMA is another method we tried to use to fill in the missing values in our data. ARIMA is usually used when time series data is concerned. While working with stationary data the ARMA (Auto Regressive Moving Average) method is used. Since weather data can be stationary or non stationary, varying from station to station, we use ARIMA model which is better suited for use with non stationary data that still has trend components. ARIMA model involves the use of a differencing component (d) which converts the data to stationary.

We used Kalman filterring along with the ARIMA method to fill missing values. It operate on state-space models of the form:

$$y_t = Z_{\alpha_t} + \epsilon_t \tag{1}$$

$$\alpha_{t1} = T_{\alpha 1} + \eta \tag{2}$$

$$\alpha_1 \sim N(\alpha_{t1}, P_1) \tag{3}$$

where yt is the observed series (with possible missing values) but ∝t is fully unobserved. The first equation (the ”measurement” equation) says that the observed data is related to the unobserved states in a particular way. The second equation (the ”transition” equation) says that the unobserved states evolve over time in a particular way.

The Kalman filter operates to find optimal estimates of ∝t (∝t is assumed to be Normal: ∝t N(at,Pt), so what the Kalman filter actually does is to compute the conditional mean and variance of the distribution for ∝t conditional on observations up to time t). In the typical case (when observations are available) the Kalman filter uses the estimate of the current state and the current observation yt to do the best it can to estimate the next state ∝t+1, as follows:

$$\alpha_{t+1} = T_{a_1} + K_t(y_t - Z_{\alpha_t}) \tag{4}$$

$$P_{t+1} = TP_t(T - K_t Z)' + Q \tag{5}$$

where Kt is the ”Kalman gain”.

When there is not an observation, the Kalman filter still wants to compute at+1 and Pt+1 in the best possible way. Since yt is unavailable, it cannot make use of the measurement equation, but it can still use the transition equation. Thus, when yt is missing, the Kalman filter instead computes:

$$\alpha_{t+1} = T_{a_t} \tag{6}$$

$$P_{t+1} = TP_tT' + Q \tag{7}$$

Essentially, it says that given αt, its guess as to αt+1 without data is just the evolution specified in the transition equation. This can be performed for any number of time periods with missing data.

If there is data yt, then the first set of filtering equations take the best guess without data, and add a "correction" in, based on how good the previous estimate was. [3]

2.4 Missing Value Imputation by Interpolation

In this method the missing value imputes by a Interpolation method which can be any of linear, Spiline or Stine Interpolation methods.

2.5 Missing Value Imputation by an Observation

In this method the missing value fills using the Last Observation Carried Forward or Next Observation Carried Backward.

2.6 Missing Value Imputation by Moving Average (MA) Methods

In this method the missing value can fills by any of Simple Moving Average, Linear Weighted Moving Average or Exponential Weighted Moving Average method.

2.7 Missing Value Imputation by Mean Value

And finally in this method we impute the missing value by simple mean value.

3 Preliminary Results

3.1 Data Integration and Reduction

As part of data preprocessing step, we have selected subsets for each continent based on the GRDC station catalogue. From the catalogue, we selected stations that belong to same WMO region, sub-region and those who had a daily missing data rate above a certain threshold. As preliminary step to selection phase, we derived the associate period of active state for each state and group the stations based on each operational year. After we got the station subset selected, we calculated the nearest stations for each of the station operational in a given year based on Haversine distance and chose 5 closest stations for each of them. Thus we were able to create more compact and reduced dataset for each continent.

3.2 AI Method performance

Using this approach, we chose 3 closest stations to the target station. The missing precipitation values of the target station are filled by the closest stations precipitation data based on last 1-2 years. Data set was divided into train and test set by 75% split. This method used Sequential model which is a linear stack of layers.Sequential model is trained with 120 epochs and adam optimizer. Actual and predicted missing values comparison are shown in figure 5.

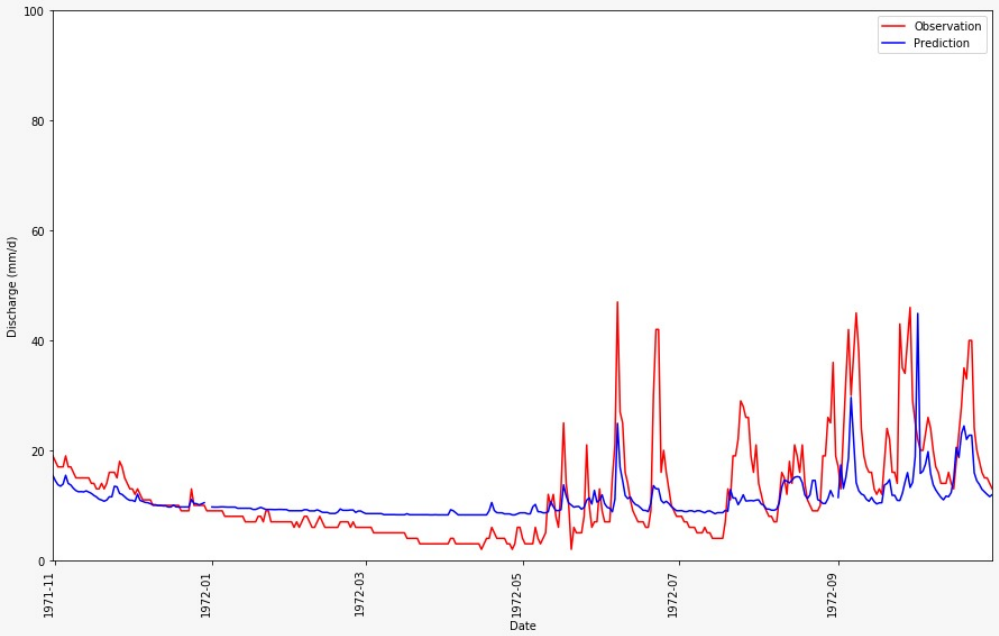

Figure 1: Missing values predicted by AI method

3.3 LSTM network Performance

To prepare the input for LSTM model, we chose 5 closest stations in every batch and observed their data trend with heatmap visualization. We observed range of years where data was most compact for those stations only. Then null values were dropped for training purposes and the rest were normalized using Min-Max scalar. Dataset were divided into train and test set by 75% split. To create the LSTM network, we needed a look back window function. We look back at 3 stations to predict the value for fourth stations,

we can say the network has 5 input for each of stations and 3 labels to predict. We trained the model for 300 epochs, batch size of one, optimized with Adam optimizer and different learning rates. RMSE value for Train set was 1.96 and 11.55 for test set. Missing value prediction results from train and test on North America stations is shown in Fig.2 and Fig.3.

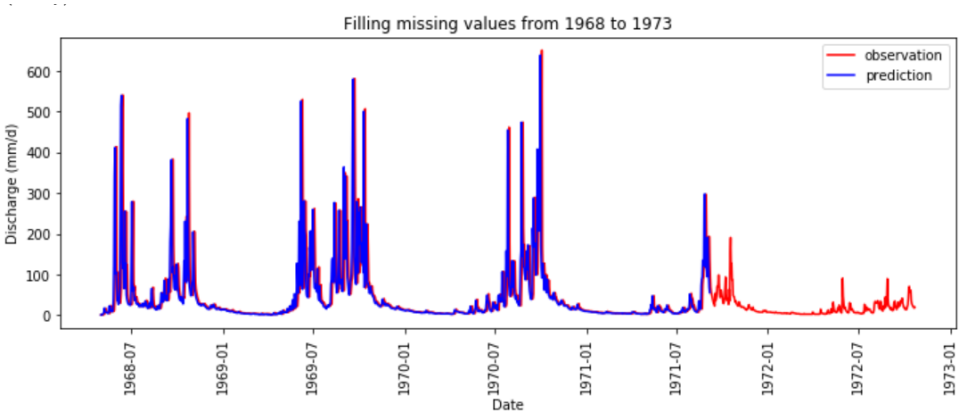

Figure 2: Train Prediction Result from LSTM network on North America stations

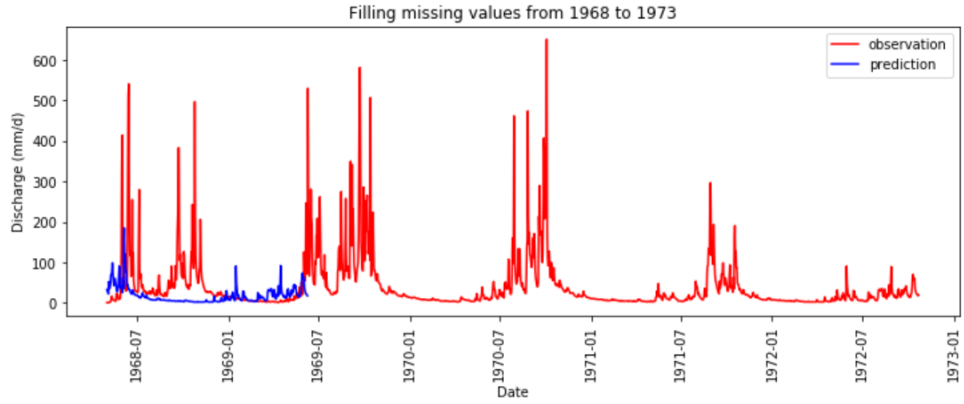

Figure 3: Test Prediction Result from LSTM network on North America stations

3.4 Performance of the Rest of the Methods

We selected a station from the North America datasets and replaced the missing values by imputation with the methods of interpolation, Kalman, Locf, Mean and Moving Average. The selected station does not have any missing values so we randomly put some of them NA to predict them with the suggested model. Thus, at the end we are able to compare our model results with the real observations that we assumed NA. Results are shown in the Figure 4 and Figures 5-9.

Date	Real Observation	ContainMissingValues	Interpolation	Kalman	Locf	Mean	MovingAverage
1860-01-01	6650	6650	6650	6650	6650	6650	6650
1860-01-02	6650	6650	6650	6650	6650	6650	6650
1860-01-03	6650	6650	6650	6650	6650	6650	6650
1860-01-04	6650	6650	6650	6650	6650	6650	6650
1860-01-05	6650	6650	6650	6650	6650	6650	6650
1860-01-06	6650	6650	6650	6650	6650	6650	6650
1860-01-07	6650	6650	6650	6650	6650	6650	6650
1860-01-08	6650	6650	6650	6650	6650	6650	6650
1860-01-09	6650	6650	6650	6650	6650	6650	6650
1860-01-10	6650	6650	6650	6650	6650	6650	6650
1860-01-11	6650	6650	6650	6650	6650	6650	6650
1860-01-12	6650	6650	6650	6650	6650	6650	6650
1860-01-13	6650	6650	6650	6650	6650	6650	6650
1860-01-14	6650	6650	6650	6650	6650	6650	6650
1860-01-15	6650	6650	6650	6650	6650	6650	6650
1860-01-16	6650	6650	6650	6650	6650	6650	6650
1860-01-17	6650	6650	6650	6650	6650	6650	6650
1860-01-18	6650	6650	6650	6650	6650	6650	6650
1860-01-19	6650	6650	6650	6650	6650	6650	6650
1860-01-20	6650	NA	6650	6646.5733	6650	5875.603	6650
1860-01-21	6650	6650	6650	6650	6650	6650	6650
1860-01-22	6650	6650	6650	6650	6650	6650	6650
1860-01-23	6650	6650	6650	6650	6650	6650	6650
1860-01-24	6650	6650	6650	6650	6650	6650	6650
1860-01-25	6650	6650	6650	6650	6650	6650	6650
1860-01-26	6650	NA	6650	6621.7191	6650	5875.603	6650

Figure 4: Missing value imputation result using rest of the methods on a sample streamflow station (Niagara River) from North America

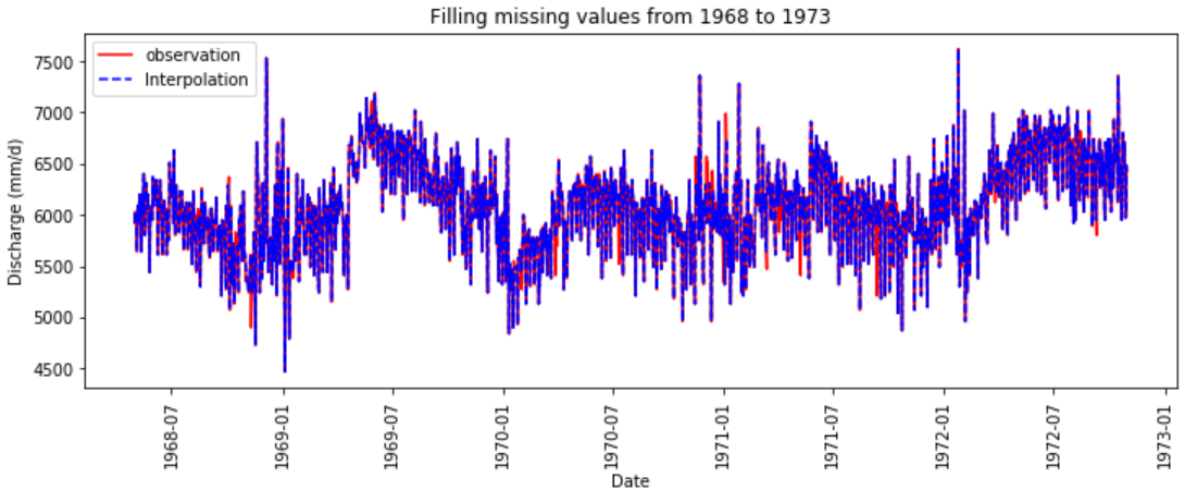

Figure 5: Missing value imputation result using Interpolation Method on a sample streamflow station (Niagara River) from North America

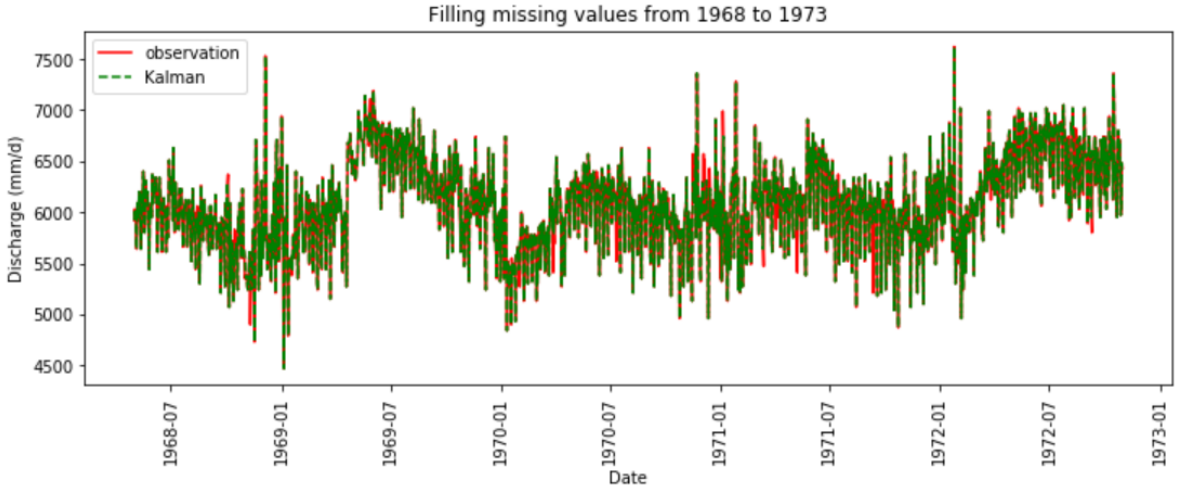

Figure 6: Missing value imputation result using Kalman Method on a sample streamflow station (Niagara River) from North America

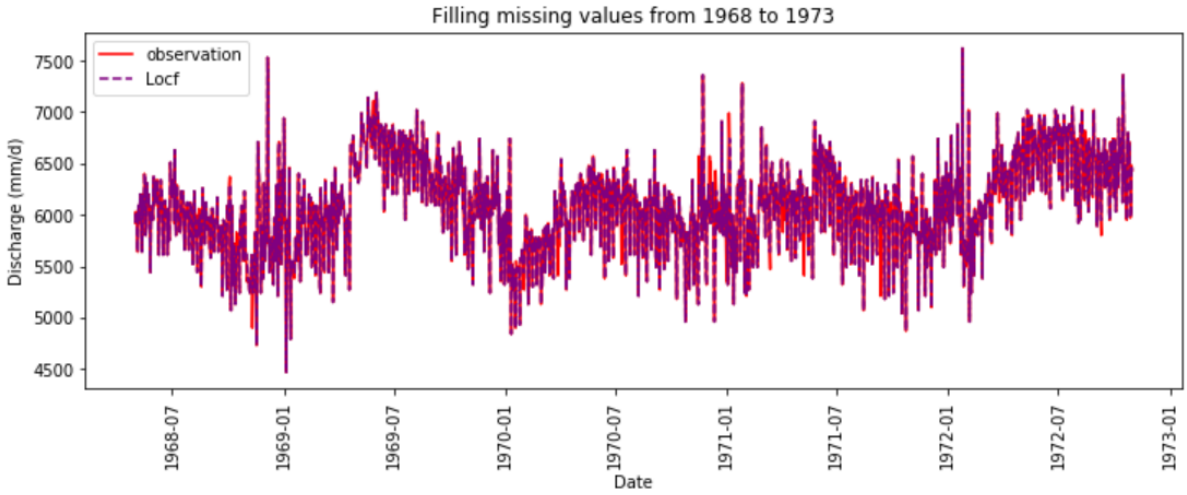

Figure 7: Missing value imputation result using Locf Method on a sample streamflow station (Niagara River) from North America

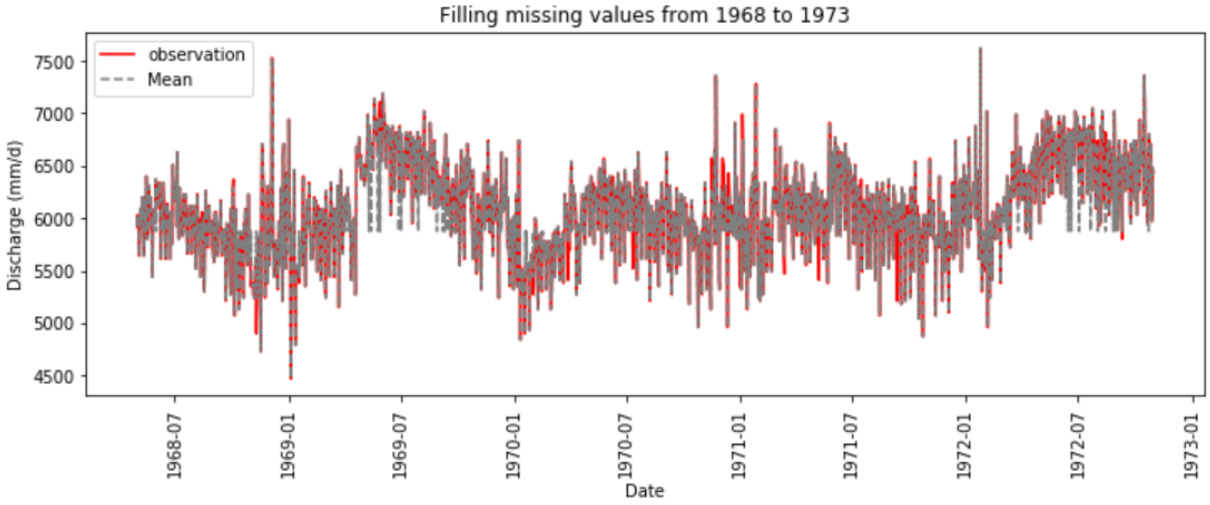

Figure 8: Missing value imputation result using simple mean method on a sample streamflow station (Niagara River) from North America

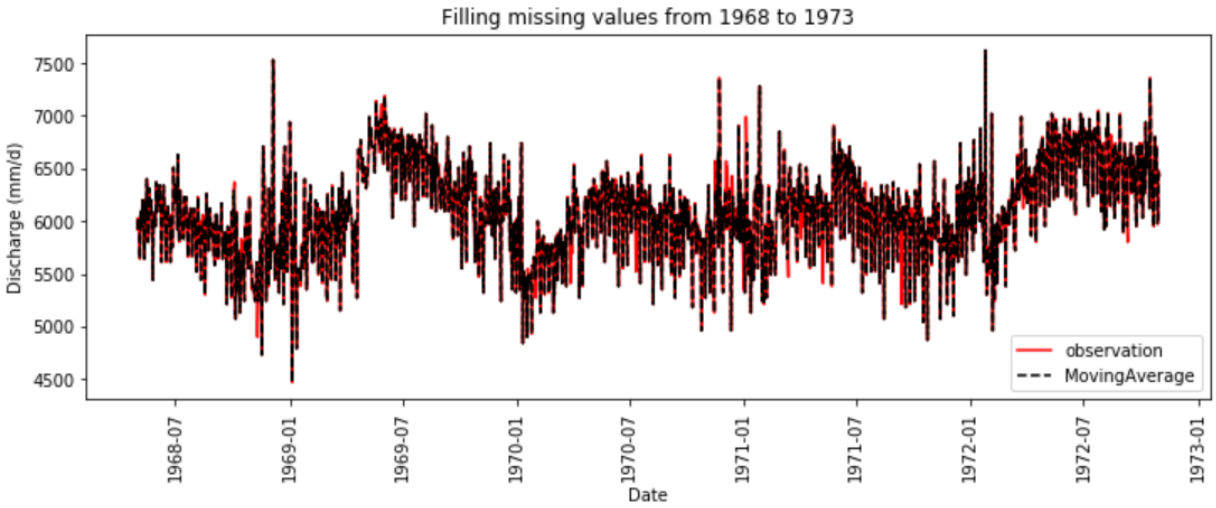

Figure 9: Missing value imputation result using Moving Average method on a sample streamflow station (Niagara River) from North America

References

[1] J. K. BEVEN. Rainfall-runoff modelling the primer, john willey & sons ltd, new york. 2000.

[2] T. K. Chang, A. Talei, S. Alaghmand, and M. P.-L. Ooi. Choice of rainfall inputs for event-based rainfall-runoff modeling in a catchment with multiple rainfall stations using data-driven techniques. *Journal of Hydrology*, 545:100–108, 2017.

[3] J. Durbin and S. Koopman. *Time series analysis by state space methods (No.38)*. Oxford University Press, 2012.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] R. J. Hyndman, Y. Khandakar, et al. *Automatic time series for forecasting: the forecast package for R*. Number 6/07. Monash University, Department of Econometrics and Business Statistics . . . , 2007.

[6] T. J. Mulvaney. On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and flood discharges in a given catchment. *Proceedings of the institution of Civil Engineers of Ireland*, 4:19–31, 1851.

[7] A. Radfar and T. D. Rockaway. Captured runoff prediction model by permeable pavements using artificial neural networks. *Journal of Infrastructure Systems*, 22(3):04016007, 2016.

[8] J. Salas, M. Markus, and A. Tokar. Streamflow forecasting based on artificial neural networks. In *Artificial neural networks in hydrology*, pages 23–51. Springer, 2000.

[9] A. S. Tokar and P. A. Johnson. Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3):232–239, 1999.

[10] Y. Yu, H. Zhang, and V. P. Singh. Forward prediction of runoff data in data-scarce basins with an improved ensemble empirical mode decomposition (eemd) model. *Water*, 10(4):388, 2018.