2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService) | 978-1-6654-5890-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigDataService55688.2022.00035

Streamflow Prediction Using a Hybrid Methodology Based on Convolutional Neural Network and Long Short-Term Memory

Juan F. Ramirez Rochac
Department of Computer Science and Information Technology
University of the District of Columbia
Washington, D.C., USA
jrochac@udc.edu

Nian Zhang
Department of Electrical and Computer Engineering
University of the District of Columbia
Washington, D.C., USA
nzhang@udc.edu

Tolessa Deksissa
Water Resources Research Institute
University of the District of Columbia
Washington, D.C., USA
tdeksissa@udc.edu

Wagdy H. Mahmoud
Department of Electrical and Computer Engineering
University of the District of Columbia
Washington, D.C., USA
wmahmoud@udc.edu

***Abstract*— This paper proposes a real-time, long short-term memory (LSTM) based low flow forecast system, while utilizing historical streamflow to make prediction of the probability of flows dropping below drought trigger levels for the Potomac River basin. The proposed recurrent neural network learns to predict the value of the next time step of the time sequence. We evaluate the prediction accuracy of the proposed LSTM-based model with real-world data and compare it to other state-of-the-art baseline models as well as other LSTM variants. The experimental results show that the prediction accuracy of the proposed method outperforms other methods. This design will help improve the performance of the decision support system for drought management.**

Keywords*— *time series prediction, water quantity prediction, long-short term memory, Scharr filtering

I. Introduction

As streamflow quantity problem become increasingly severe, accurate prediction and effective management of scarcer water resources will become critical. In addition, river streamflow prediction is one of most complicated and difficult hydrological problems because the physical processes that control it are too complex to allow adequate description by a system of appropriate equations. It is also because of the insufficient knowledge on the driving factors and their impact on streamflow, as well as the lack of reliable prediction and design methodologies. Noise injected by low quality sensors, worn out equipment, and changes in the environment also impact this process. Therefore, understanding and accurate evaluations of streamflow quantity are critical to enhance the performance of an assessment operation and develop better water resources management and plan.

Recently, most research studies have applied some form of machine learning/deep learning (ML/DL) model to accurately predict streamflow. With DL approaches at the current state-of-the-art, research is shifting to further explore the benefits of hybrid streamflow prediction models. However, new models are needed to deal with the nonlinearity and noise that affects streamflow prediction. The main contribution of this paper is a robust approach to deal with noise in water quantity prediction: (i) to adopt the Scharr filter to reduce noise in time series datasets; (ii) to improve robustness in LSTM-based models; (iii) to conduct extensive experiments using real-world streamflow data.

II. Methodology

The overview of our approach is shown in Fig. 1, and each component is further detailed in the following sub-sections.

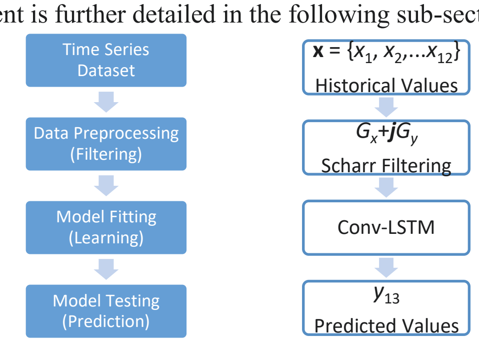

Fig. 1. Overview of the proposed framework for water quantity prediction.

A. Time Series Dataset

The real-world dataset consists of monthly adjusted river streamflow in cubic feet per second collected at the Potomac Basin above Little Falls, which is near Washington DC, USA from 1930 to 2021. The values were collected by the National Water Information System (NWIS) at the U.S. Geological Survey, Station No. 01646502. On the other hand, the synthetic noise-augmented data was generated by using an Additive Gaussian White Noise (AGWN) with a signal-to-noise ratio (SNR) equal to 20. The noise-augmented dataset is the element-wise addition of the real-world streamflow values plus the synthetic noise values.

B. Data Pre-processing

In this subsection, we introduce the Scharr filtering for time series data. And, we also describe the data partition for learning and prediction. To filter out noise and outliers in time series data, we use the complex Scharr operator as defined by the mask in Eq. (1).

$$G_x + jG_y = \begin{bmatrix} -3-3j & 0-10j & +3-3j \\ -10+0j & 0+0j & +10+0j \\ -3+3j & 0+10j & +3+3j \end{bmatrix} \tag{1}$$

DOI 10.1109/BigDataService55688.2022.00035

Where *Gx* is the horizontal gradient and *Gy* is the vertical gradient. This complex operator uses the real part for *Gx* and the imaginary part for *Gy*.

C. *Proposed Deep Learning Models*

In this subsection, we present six LSTM-based variants for time series prediction. First, in Fig. 2, we have the vanilla variants, Single LSTM and Stacked LSTM which consist of vanilla LSTM Layers. Then, in Fig. 3, we have the bidirectional variants, Bi+LSTM and Stacked Bi+LSTM which consists of bidirectional layers. And then, in Fig. 4, we have the convolutional variants, CNN+LSTM and ConvLSTM models. All variants take previous x_i and predict the next y.

Fig. 2. LSTM models: The left inset (a) depicts a single LSTM-layer model in gold while the right inset (b) shows a stacked LSTM model. Both use a dense layer in grey and a rectified linear (ReLU) activation function.

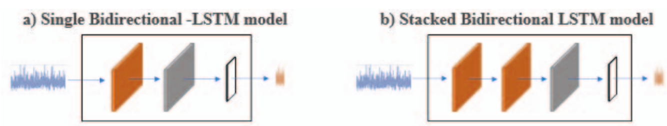

Fig. 3. Bidirectional LSTM models: The left inset (a) depicts a single BiLSTM layer model in maroon while the right inset (b) shows a stacked bidirectional LSTM model. Both models use a dense layer and a ReLU.

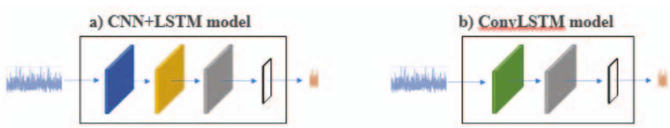

Fig. 4. Convolutional LSTM models: The left inset (a) depicts a CNN layer in blue and a LSTM layer in gold while the right inset (b) shows a ConvLSTM layer in green. Both use a dense layer in grey and a ReLU.

D. *Model Testing*

All six variants were tested using 10% of the dataset. To partition the dataset into training set and test set, we use a 9:1 ratio. Thus, 90% of the input dataset is used for fitting the model while the remaining 10% is used for prediction. For comparison purposes, we use root-mean-square error (RMSE) to measure accuracy.

III. Experimental Results

A. *Experimental Setup*

To evaluate our proposed framework in terms of accuracy and resistance to noise, two pipelines are implemented. A pipeline for original data with and without a filter and another with augmented data with and without the filter. In all the experimental settings, the proposed LSTM-variants use the Adam optimizer and a learning rate of 0.00001 for 200 epochs. The test bench for all performance experiments was implemented using Python 3.7. The runtime environment consists of a 2-core CPU, Intel(R) Xeon(R) @ 2.20GHz, a single GPU, Tesla K80, 4992 CUDA, 12GB GDDR5, 13GB of RAM and 80GB of HDD on a Linux-based virtual machine.

B. *Comparative Analysis*

We evaluated out-of-box ARIMA-based models using a walk-forward validation. Then, we evaluated two least squares support vector regressor (LSSVR) models, such as, linear kernel and radial kernel. LSSVR showed comparable performance to the proposed GA-based model. We also present direct and hybrid LSTM-based models. Regarding the hybrid models, Variational Mode Decomposition (VMD) is used to preprocess the streamflow time series data and deep learning LSTM-based models to forecast future streamflow values. Finally, the proposed Scharr+ConvLSTM model is presented. All the experimental results are compiled into Table 1.

Table 1 RMSE Comparison of Prediction Models.

Models	**RMSE**	
	Original Data	**Augmented Data**
AR *order*=(2, 1, 0)	0.080	0.085
MA *order*=(0, 1, 2)	0.081	0.087
ARIMA *order*=(2,1,2)	0.076	0.081
SARIMAX *order*=(1,1,1), *seasonal_order*=(1,1,0,12)	0.089	0.094
LSSVR *C*=200, *kernel*=linear	0.106	0.111
LSSVR *C*=200, *kernel*=rbf, *gamma*=0.0625	0.096	0.101
LSTM *hidden layers*=200	0.108	0.0052
BiLSTM *hidden layers*=200	0.118	0.0049
ConvLSTM *hidden layers*=256	0.106	0.0048
VMD + LSTM *modes*=18	0.055	0.056
VMD + BiLSTM *modes*=18	0.059	0.064
VMD + ConvLSTM *modes*=18	0.056	0.067
Scharr + LSTM	0.0036	0.0038
Scharr + BiLSTM	0.0038	0.0035
Scharr + ConvLSTM	0.0030	0.0031

IV. Conclusions

Our experiments suggest that the proposed model improves the overall prediction accuracy on both real-world data and noise-augmented data when compared to other LSTM-based models and to other baseline models. In addition, our proposed model achieves comparable performance in terms of prediction time. In the future, we will focus on multivariate LSTM-based models, explore additional filters and employ more performance metrics such as R-squared.

Acknowledgment

This work was supported in part by NSF grant #2011927, DoD grant #W911NF1810475, and USGS grant #2020DC142B.