Article:

Zhang, J, Chen, X orcid.org/0000-0002-2053-2448, Khan, A orcid.org/0000-0002-7521-5458 et al. (5 more authors) (2021) Daily runoff forecasting by deep recursive neural network. Journal of Hydrology, 596. 126067. ISSN 0022-1694

https://doi.org/10.1016/j.jhydrol.2021.126067

Daily runoff forecasting by deep recursive neural network

Jiangwei Zhang[1,2]; Xiaohui Chen [1*]; Amirul Khan [1]; You-kuan Zhang [2]; Xingxing Kuang [2]; Xiuyu Liang [2]; Maria L Taccari[1]; Jonathan Nuttall[3]

[1]School of Civil Engineering, University of Leeds, Leeds, LS2 9JT, UK

[2]School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, PR China

[3]Deltares, Delft, The Netherlands

Corresponding author:X Chen (X.Chen@leeds.ac.uk)

Highlights:

- A new deep RNN model was developed to predicate highly nonlinear daily runoff.
- The selection of input variables has a huge impact on deep RNN forecasting results.
- PCA method was applied to improve the accuracy of deep RNN Model.

Abstract

In recent years deep Recurrent Neural Network (RNN) has been applied to predict daily runoff, as its ability of dealing with the high nonlinear interactions among the complex hydrology factors. However, most of the existing studies focused on the model structure and the computational load, without considering the impact from the selection of multiple input variables on the model prediction. This article presents a study to evaluate this influence, and provides a method of identifying the best meteorological input variables for a run off model. Rainfall and multiple meteorological data has been considered as input to the model. Principal Component Analysis (PCA) has been applied to the data as a contrast, to reduce dimensionality and redundancy within this input data. Two different deep RNN models, a long-short term memory (LSTM) model and a gated recurrent unit (GRU) model, have been comparatively applied to predict runoff with these inputs. In this study, the Muskegon river and the Pearl river were taken as examples. The results demonstrate that the selection of input variables have a significant influence on the predictions made using the RNN while the RNN model with multiple meteorological input data is shown to achieve higher accuracy than rainfall data alone. PCA method can improve the accuracy of deep RNN model effectively as it can reflect core information by classifying the original data information into several comprehensive variables.

Keywords: runoff forecasting; deep learning; recursive neural network (RNN); long-

short term memory (LSTM); gate recurrent unit (GRU); principal component analysis (PCA)

Abbreviations:

RNN, recursive neural network;

PCA, Principal component analysis;

LSTM, long-short term memory;

GRU, gated recurrent unit;

ANN, artificial neural network;

AI, artificial intelligent;

SVM, support vector machine;

ARIMA, autoregressive integrated moving average;

AR, autoregressive;

ARMA, auto-regressive moving average;

RMSE, the root mean square error;

NSE, Nash-Sutcliffe Efficiency;

R^2, the coefficient of determination;

MAE, the mean absolute error

WMAPE, weighted mean absolute percentage error

1. Introduction

Water resources are significant for sustainable development of economics and ecology and runoff forecasting plays a significant role in water resources planning and management, for example, flood control, dam planning and reservoir operation (Napolitano et al. 2011; Yuan et al. 2018). However, rain-runoff forecasting is a difficult issue in hydrological process simulation, because of the highly non-linear behaviour of the factors governing the hydrology system in the space-time domain (Wang et al. 2009; Zhu et al. 2016). In the past decades, a great deal of effort has been devoted to runoff prediction (Yuan et al. 2018). Generally, the existing methods can be divided into process-driven methods and data-driven methods.

Process-driven methods are based on partial differential equation and linear equation which incorporate the physical processes (Bittelli et al. 2010; Partington et al. 2012). However, runoff is affected by many uncertainties, high complexity, non-stationarity, dynamism and non-linear factors, so it is difficult to forecast runoff by process-driven method accurately(Yoon et al. 2011). Data-driven methods are also used for rain-runoff forecasting. Basically, data-driven methods are statistical methods which just focus on the input-output relationship without explicit causality between factors in a specific system (Solomatine&Ostfeld 2008). Some researchers assumed the input-output relationship is linear and predicted the runoff by methods like the autoregressive integrated moving average (ARIMA) model (Valipour 2015), autoregressive (AR) model (Pulukuri et al. 2018), and auto-regressive moving average (ARMA) model (Mehdizadeh et al. 2019). Other researchers believe the forecast result can be improved by considering the non-linear characteristics hidden in runoff series(Amiri 2015). Machine learning methods like artificial neural network (ANN)(Riad et al. 2004) and support vector machine (SVM)(Li et al. 2014) are widely used for runoff prediction. However, one of the limitations of these machine learning methods is that they will struggle when system behaviour is dominated by spatial or temporal context. (Reichstein et al. 2019; Zhongrun&Ibrahim 2020). Meanwhile, the simple structure of these data-driven models limits their ability to address the high nonlinear relationship between the weather time series data and runoff time series data (Amiri 2015).

Deep learning (DL) is a new hot topic in mechanical learning. The most distinguishing characteristic of deep networks is multiple neuron layers in neural network architecture, which provides a higher ability to represent complex functions than non-deep neural network (Raghu et al. 2017). DL method has been widely used in language modelling (Magassouba et al. 2020), image analysis (Litjens et al. 2017), recommendation systems (Ji et al. 2013) and other fields.

Recently, researchers have paid attention to RNNs and its variants (such as LSTMs and GRUs) and have found that deep RNNs have better performance for runoff time-series prediction. Xiang (2020) applied a runoff prediction model based on LSTM and the seq2seq structure to estimate runoff for next 24 hours. Kratzert (2018) employed the LSTM model in catchments with snow and compared the model prediction result with Soil Moisture Accounting Model (SAC-SMA) coupled with the Snow-17 snow routine. Wang (2020) used RNN to perform meteorological statistical downscaling and evaluate the hydrological response to the downscaled meteorological data by SWAT. Meanwhile, many researches focused on the time-steps in RNN models, Kao (2020) proposed a Long Short-Term Memory based Encoder-Decoder (LSTM-ED) model for multi-step-ahead flood forecasting and compared the performance of the FFNN- and LSTM-based ED models with different time steps; Chen(2020) and Cheng(2020) evaluated the importance of long lead-time and short lag-time in LSTM. The results prove the advantage of deep learning model. According to the input variables, the Rainfall-Runoff studies can be divided into 2 categories:(1) Some researchers employ rainfall data as input to predict runoff (Hu et al. 2018; Le et al. 2019); (2) Other researchers prefer to incorporate rainfall data with multiple meteorological as input parameters, enabling various factors related to runoff to be considered within the model (de la Fuente et al. 2019). However, limited research has been focused on the influence of different input variables on rain-runoff forecasting by using deep RNN.

In order to consider the impact from the selection of input variables on the model prediction and find a way to improve accuracy when multiple input variables were employed, this study investigates the performance of deep RNN models on runoff forecasting with different input variables and an optimized input is identified based on the PCA method. To make the result more credible, two USGS stations in different climatic zones were chosen as study areas. A general description of different algorithms and data source is provided in Section 2. While the predicting results with different input are discussed in Section 3. Conclusions are presented in the last part of the manuscript.

2. Method and data

2.1. Data collection

To cover diverse hydroclimatological regimes, the Muskegon River and the Pearl River were chosen as the study area. As shown in Fig 1, The Muskegon River is located at the west of Michigan, U.S. state and it belongs to temperate continental climate. The river comes from Houghton Lake and flows southwest to Muskegon Lake stretching nearly 384km. Muskegon River basin is nearly 6,100 km^2 and is composed of 40 subwatersheds (Ray et al. 2010). Muskegon River plays a crucial role in the social economy and natural ecology of the basin. Rogers dam, Hardy dam and Croton dam on Muskegon River provide nearly 23000 people with a cleaner source of electricity. The runoff of Muskegon River influence ecosystem and biogeochemistry in Lake Michigan (Johengen et al. 2008).

The Pearl River is in southern Mississippi, U.S. state which belongs to humid subtropical climate. The Pearl River run from Neshoba County and flow to Lake Borgne with the length of length of 715 km (Taylor&Grace 1995). The Ross Barnett Reservoir is the most important water facility which provides drinking water for residents in Metropolitan Jackson.

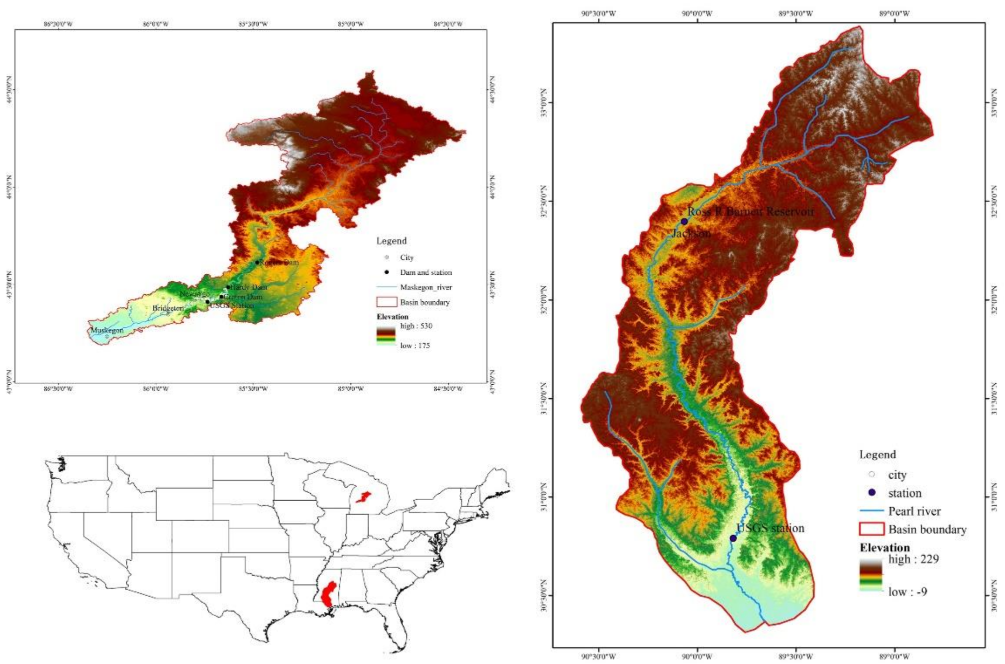

Figure 1 Overview of two study areas.

Daily runoff time series data was gathered from USGS Hydrological station 02489500 and 04121970. Daily meteorologicaltime-series data were collected from Weather Underground and NOAA. The meteorological data includes the following data (Table

1):

Table 1 Details of the meteorological data in the Muskegon River and Pearl River

	meteorological time series data in the Muskegon River	meteorological time series data in the Pearl River
Indexes	● Max-temperature (°C), ● Mean-temperature (°C), ● Min-temperature (°C), ● Max-dew point (°C), ● Mean-dew point (°C), ● Min-dew point (°C), ● Max-humidity (%), ● Mean-humidity (%), ● Min-humidity (%), ● Max-sea level pressure (hPa), ● Mean-sea level pressure (hPa), ● Min-sea level pressure (hPa), ● Max-windspeed (km/h),	● Max-temperature (°C), ● Mean-temperature (°C), ● Min-temperature (°C), ● Max-dew point (°C), ● Mean-dew point (°C), ● Min-dew point (°C), ● Max-humidity (%), ● Mean-humidity (%), ● Min-humidity (%), ● Max-sea level pressure (hPa), ● Mean-sea level pressure (hPa), ● Min-sea level pressure (hPa), ● Max-windspeed (km/h),

	● Mean-windspeed (km/h), ● Min-visibility (km), ● Max-visibility (km), ● Mean-visibility (km), ● Precipitation (mm).	● Mean-windspeed (km/h), ● Precipitation (mm).
Duration	01/10/1995-01/01/2020	01/01/2000-01/01/2020

2.2 Data pre-processing

Data pre-processing consists of data division and data cleaning. In addition, as 18 indicators of data were collected, PCA was employed to reduce the dimensionality of the input data, to provide an alternate input dataset (See below.)

Data cleansing: The missing or the outlying points in time series data reduce accuracy and quality of training and prediction. To discount the influence, the outlying data were identified by 6σ rule which assumes data outside the range of $\overline{D} \pm 6\sigma$ is outlying data (Jeong et al. 2017), where $\overline{D}$ and σ are the average value and the standard deviations of time series data respectively. The outlying points and vacancy were replaced by average value of the same date in different years.

To avoid the influence of dimension on the training process, data were normalized into a standardized range by the following equation:

$$D_N = \frac{D-\bar{D}}{\sigma(D)} \tag{1}$$

Data division: To prevent overfitting and test the predictive capabilities of the model, the data was divided into two parts: 80% was used to train the suggested models, and the other 20% was used to test the trained models. In process-driven modelling, the value of past history is tied to the time lag between input and output response, such as the rainfall and the ground water table response. However, in data-driven modelling, the past history is just a hyperparameter which is not directly related to the physical behaviour (Jeong&Park 2019). So, in the training and testing part, the value of history was identified by trial-and -error.

The PCA method: The PCA method extracts several principal comprehensive variables form original data by covariance matrix, to persist core information and eliminate noise. PCA has been widely used in the literature and data mining since its introduction by Pearson (1901). The calculation processes of PCA method are as following (Hotelling 1933):

1) Processing the normalized data as matrix D_N;
2) Calculating the correlation coefficient matrix Ccm based on the matrix D_N as:

$$Ccm_{ij} = Cov(D_{N_i}, D_{N_j}) \tag{2}$$

where D_{N_i} is the indicator vector in matrix D_N.

3) Calculating feature values λ and feature vectors of matrix CCM and put feature values in orderas $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.

4) Calculating the principal component contribution rate Ccr_i and the cumulative contribution rate Ccr:

$$Ccr_i = \frac{\lambda_i}{\sum_{k=1}^{n} \lambda_k} \tag{3}$$

$$CCR = \frac{\sum_{p=1}^{i} \lambda_p}{\sum_{k=1}^{n} \lambda_k} \tag{4}$$

5) When the cumulative contribution rate was greater than 85%, the number of principal components can be determined. Processing the respective feature vectors as matrix M, the data after dimension reduction X_{PCA} can be calculated as:

$$X_{PCA} = MX \tag{5}$$

2.3 Baseline model

In data-driven model research, it is recommended to use some simple but effective forecasting method as baseline model to provide benchmarks (Hyndman&Athanasopoulos 2018). In this article, ridge regression is employed as the baseline model. Ridge regression (Hoerl 1959) is an effective method which is widely used in machine learning and hydrology (Chen et al. 2018; Miche et al. 2020). Based on linear regression, an L2 regularization term is applied in the loss function of ridge regression. By this way, ridge regression gains a better ability of generalization. To make the baseline model concise, rainfall data is input as it is the common variables in different inputs considered in this paper.

2.4 RNNs

RNNs consist of the input layer, hidden layer (or hidden layers) and the output layer, but different from other ANNs (artificial neural network), RNNs have the fabulous memory ability as these networks introduce state variables to store past information, and then determine the current outputs, together with the current inputs.

The RNNs model can be trained by the BPTT (Back Propagation Through Time) method which calculates not only the gradient of the cost corresponding to the input weights but also the gradient of the cost corresponding to the hidden weights of the previous time steps. However, the error of partial derivative accumulates through time steps in the BPTT method. Meanwhile, when the time step T is large, the gradient will either get very small and vanish, or get very large and explode. This problem is commonly known as the vanishing/exploding gradient problem. In recent years, the hidden block in RNNs is replaced by LSTM block or GRU block to combat vanishing/exploding grad.

2.4.1 LSTM

LSTM neural network (Hochreiter&Schmidhuber 1997) replaces the hidden block in RNNs with three logic gates and a memory cell as it is shown in Fig 2.

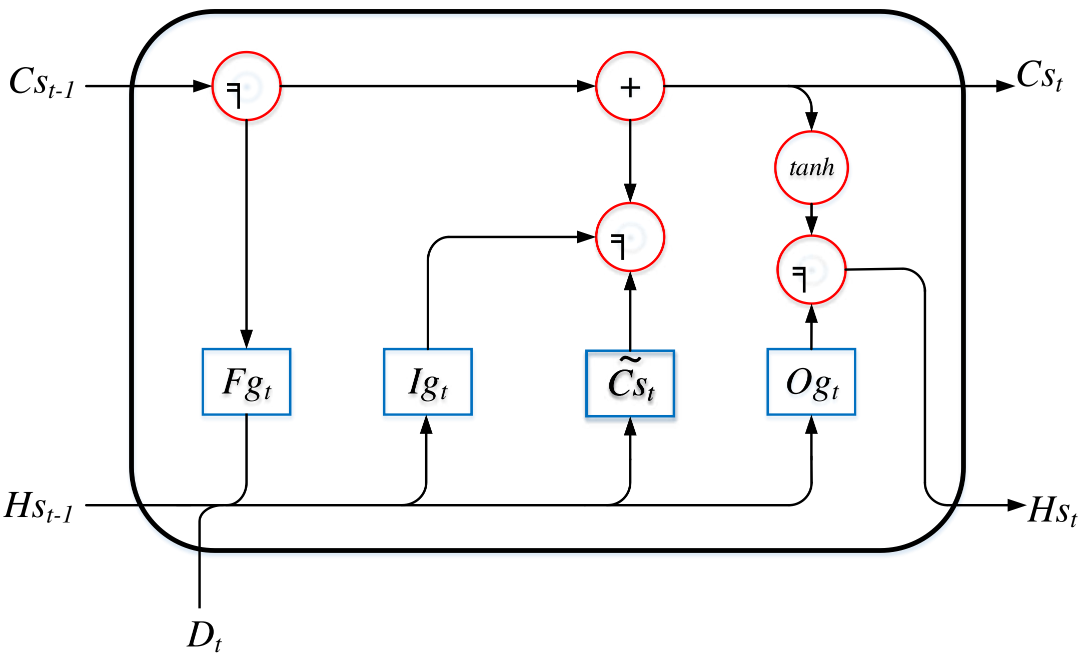

Figure 2 Neuron in the hidden layer of LSTM at time step t

In the training process, the memory cell state Cs_t and hidden state Hs_t would be updated selectively based on the input gate Ig_t and output gate Og_t. The irrelevant information in long-term memory would be forgotten by forget gate Fg_t.The hidden block of LSTM neural network can be represented as following (Amiri 2015):

Input gate:

$$Ig_t = \sigma(D_t W_{xi} + Hs_{t-1} W_{hi} + b_i) \tag{6}$$

Forget gate:

$$Fg_t = \sigma\left(D_t W_{xf} + Hs_{t-1} W_{hf} + b_f\right) \tag{7}$$

Output gate:

$$Og_t = \sigma(D_t W_{xo} + Hs_{t-1} W_{ho} + b_o) \tag{8}$$

Cell state:

$$Cs_t = Fg_t \odot Cs_{t-1} + Ig_t \odot \tilde{Cs}_t \tag{9}$$

$$\tilde{Cs}_t = tanh(D_t W_{xc} + Hs_{t-1} W_{hc} + b_c) \tag{10}$$

Hidden state:

$$Hs_t = Og_t \odot tanh(\,Cs_t) \tag{11}$$

where σ is sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, which can be used as the activation function in this step to transform input to the range of 0-1; $W_{xi}, W_{xf}\ , W_{xo}, W_{xc} \in R^{d\times h}, W_{hi}\ , W_{hf}, W_{ho}\ , W_{hc} \in R^{h\times h}$ are weight matrixes;and$b_i, b_f, b_o, b_c \in R^{1\times h}$ are biases. $\odot$ is the Hadamard product of two matrixes. The activation function in this step is $tanh$ which can ensure hidden states range from -1 to1.

2.4.2 GRU

The GRU(Cho et al. 2014; Chung et al. 2014)neural network is similar to the LSTM neural network. It replaces the hidden block in RNNs with two logic gates and the candidate hidden state $\tilde{Hs}_t$ as it is shown in Fig 3.

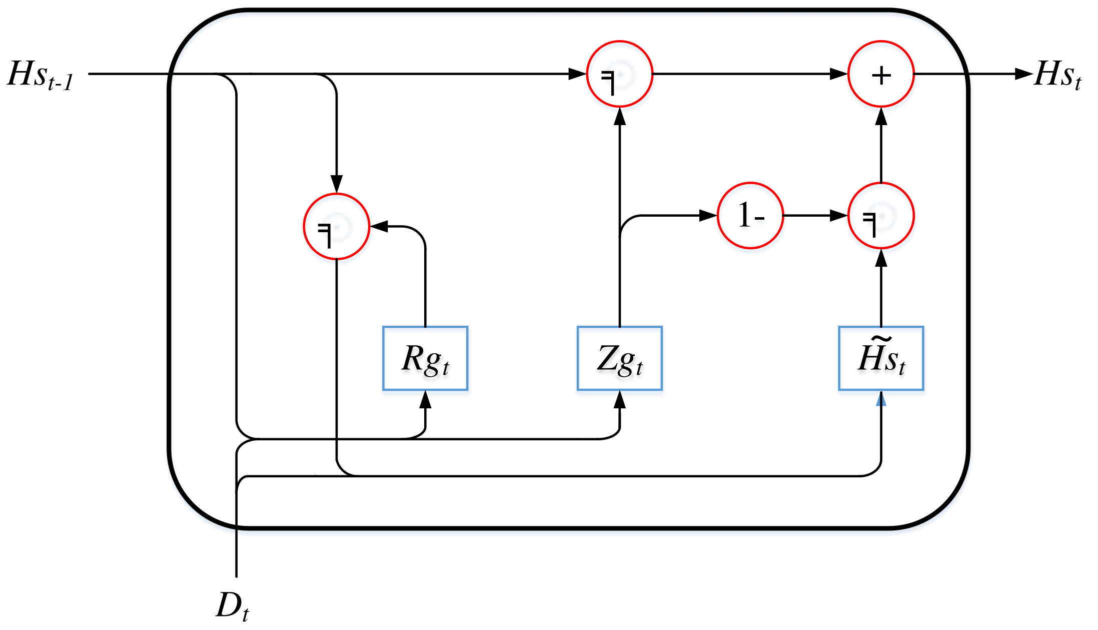

Figure 3 Neuron in the hidden layer of GRU at time step t

The hidden block of GRU neural network can be represented as following(Cho et al. 2014; Chung et al. 2014):

Reset gate：

$$Rg_t = \sigma(D_t W_{xr} + Hs_{t-1} W_{hr} + b_r) \tag{12}$$

Update gate：

$$Zg_t = \sigma(D_t W_{xz} + Hs_{t-1} W_{hz} + b_z) \tag{13}$$

Candidate hidden state：

$$\tilde{Hs_t} = tanh(D_t W_{xh} + (Rg_t \odot H_{t-1}) W_{hh} + b_h) \tag{14}$$

Hidden state:

$$Hs_t = Zg_t \odot Hs_{t-1} + (1 - Zg_t) \odot \tilde{Hs_t} \tag{15}$$

where $W_{xr}, W_{xz}, W_{xh} \in R^{d \times h}$ and $W_{hr}, W_{hz}, W_{hh} \in R^{h \times h}$ are weight matrix;$b_r, b_z, b_h \in R^{1 \times h}$are biases.

The update gate Zg_t is used to capture long-term dependencies in time series. Meanwhile, the reset gate Rg_t and the candidate hidden state are used to learn the short-term dependencies in time series. The candidate hidden state presents the influence of the previous hidden state on present hidden state. If the elements in the reset gate are close to 1, the hidden state of the previous hidden state will be reserved.

If the elements in the reset gate are close to 0, the hidden state of the previous hidden state will be forgotten.

2.4.3 Dropout

Deep RNN is an effective method to deal with big data due to its memory ability. However, it would be overfitting when the input is high-dimensional. Dropout is a regularization method and provides an effective solution for this problem (Srivastava et al. 2014).

The main idea of dropout is that there is a certain probability that every neuron in a certain layer where the dropout method is applied will not be updated during each training iteration. By this way, the output will not be overly dependent on some elements of hidden layer. However, due to the memory ability of RNNs, the dropout method can only be applied to no-recurrent connection between layers.

2.5Model evaluation criteria

The root means quare error (RMSE), Nash-Sutcliffe Efficiency (NSE), the coefficient of determination (R^2), the mean absolute error (MAE) and the weighted mean absolute percentage error (WMAPE) are used to evaluate the model performance.

3. Results and discussion

3.1 Data pre-processing

Time series datasets were normalized by their mean and standard deviation and the SPSS 21 was used to identify the principal component. The correlation coefficient matrix is shown in Table 2.

Table 2a The correlation coefficient matrix of multiple meteorological data in Muskegon River

	MAXTEMP	MEANTEMP	MAXDEW	MINTEMP	…	MEANVIS	PRECI	
MAXTEMP	1.00	0.98	0.94	0.93	…	-0.26	0.04	MAXTEMP
MEANTEMP	0.98	1.00	0.96	0.98	…	-0.21	0.07	MEANTEMP
MAXDEW	0.94	0.96	1.00	0.94	…	-0.18	0.16	MAXDEW
MINTEMP	0.93	0.98	0.94	1.00	…	-0.16	0.10	MINTEMP
MEANDEW	0.93	0.96	0.99	0.96	…	-0.22	0.15	MEANDEW
MINDEW	0.91	0.95	0.95	0.96	…	-0.25	0.12	MINDEW
MAVHUM	0.15	0.15	0.34	0.17	…	-0.26	0.21	MAVHUM
MEANHUM	-0.13	-0.06	0.16	0.02	…	-0.04	0.32	MEANHUM
MINHUM	-0.30	-0.21	-0.02	-0.10	…	0.14	0.28	MINHUM
MAXSEA	-0.35	-0.40	-0.44	-0.44	…	-0.14	-0.19	MAXSEA
MEANSEA	-0.20	-0.25	-0.33	-0.29	…	-0.31	-0.26	MEANSEA
MINSEA	-0.07	-0.11	-0.21	-0.15	…	-0.41	-0.29	MINSEA

MAXWIND	0.04	0.03	0.00	0.01	…	0.01	0.01	MAXWIND
MEANWIND	0.31	0.28	0.14	0.23	…	-0.09	-0.29	MEANWIND
MINVIS	0.24	0.20	0.03	0.14	…	-0.09	-0.34	MINVIS

Table 2b The correlation coefficient matrix of multiple meteorological data in Pearl River

	MAXTEM	MEANTEM	MINTEM	MAXDEW	MEANDEW	…	MINSEA	PRE
MAXTEM	1.00	0.96	0.86	0.87	0.87	…	-0.38	-0.10
MEANTEM	0.96	1.00	0.94	0.93	0.94	…	-0.44	-0.04
MINTEM	0.86	0.94	1.00	0.91	0.93	…	-0.43	0.02
MAXDEW	0.87	0.93	0.91	1.00	0.97	…	-0.53	0.08
MEANDEW	0.87	0.94	0.93	0.97	1.00	…	-0.50	0.05
MINDEW	0.80	0.87	0.91	0.87	0.93	…	-0.41	0.03
MAXHUM	0.19	0.21	0.24	0.41	0.42	…	-0.29	0.20
MEANHYM	0.04	0.14	0.27	0.42	0.45	…	-0.33	0.30
MINHUM	-0.05	0.11	0.29	0.35	0.38	…	-0.29	0.28
MAXWIND	-0.10	-0.05	-0.01	0.05	0.01	…	-0.27	0.16
MEANWIND	-0.27	-0.18	-0.12	-0.08	-0.13	…	-0.19	0.15
MAXSEA	-0.57	-0.62	-0.60	-0.64	-0.65	…	0.87	-0.13
MEANSEA	-0.47	-0.53	-0.51	-0.59	-0.58	…	0.93	-0.19
MINSEA	-0.38	-0.44	-0.43	-0.53	-0.50	…	1.00	-0.22

PRE	-0.10	-0.04	0.02	0.08	0.05	…	-0.22	1.00

The eigenvalues, variance contribution rates and cumulative variance contribution rates of the correlation coefficient matrix is shown in Table 3. 5 components in each study areas were extracted and the cumulative variance contribution rates are 92.102% and 91.84%

Table 3 The Eigenvalues and variance contribution rates

Muskegon River				Pearl River			
Ingredient	Eigenvalues	% of Variance	Cumulative %	Ingredient	Eigenvalues	% of Variance	Cumulative %
1	6.41	35.63	35.63	1	7.27	48.49	48.49
2	4.35	24.19	59.82	2	2.64	17.62	66.11
3	2.63	14.61	74.43	3	1.82	12.11	78.23
4	1.16	6.42	80.84	4	1.23	8.23	86.46
5	1.04	5.78	86.62	5	0.81	5.39	91.84
6	0.74	4.13	90.76	6	0.54	3.58	95.42
7	0.61	3.39	94.14	7	0.21	1.39	96.81
8	0.45	2.48	96.62	8	0.14	0.90	97.71
9	0.18	1.00	97.62	9	0.09	0.599	98.30
10	0.16	0.89	98.51	10	0.08	0.56	98.86

11	0.13	0.70	99.22	11	0.06	0.41	99.26
12	0.05	0.26	99.48	12	0.06	0.39	99.66
13	0.04	0.21	99.69	13	0.03	0.20	99.85
14	0.02	0.13	99.82	14	0.02	0.12	99.97
15	0.01	0.08	99.90	15	0.01	0.03	100.00
16	0.01	0.07	99.96				
17	0.00	0.03	99.99				
18	0.00	0.01	100.00				

The principal component matrix of time series dataset is shown in Table 4.

Table 4 The principal component matrix of time series dataset

Muskegon River						Pearl River					
Indicator	1	2	3	4	5	Indicator	1	2	3	4	5
MAXTEMP	0.92	0.33	-0.01	0.10	-0.03	MAXTEM	0.83	-0.49	-0.12	0.10	0.07
MEANTEMP	0.95	0.26	-0.02	0.12	-0.04	MEANTEM	0.90	-0.37	-0.10	0.16	0.08
MAXDEW	0.97	0.06	0.06	0.13	0.01	MINTEM	0.90	-0.24	-0.02	0.23	0.08
MINTEMP	0.95	0.18	-0.01	0.13	-0.04	MAXDEW	0.95	-0.11	0.03	0.17	0.04
MEANDEW	0.98	0.07	0.10	0.11	0.00	MEANDEW	0.96	-0.13	0.08	0.17	0.02
MINDEW	0.96	0.09	0.13	0.09	-0.01	MINDEW	0.90	-0.17	0.12	0.21	0.03
MAVHUM	0.35	-0.51	0.55	-0.12	0.19	MAXHUM	0.46	0.35	0.59	-0.16	-0.17

MEANHUM	0.18	-0.78	0.48	-0.02	0.13	MEANHYM	0.48	0.62	0.57	0.05	-0.13
MINHUM	0.01	-0.76	0.30	0.04	0.07	MINHUM	0.42	0.64	0.41	0.22	-0.09
MAXSEA	-0.59	0.45	0.41	0.45	0.09	MAXWIND	0.06	0.61	-0.53	0.42	-0.06
MEANSEA	-0.46	0.61	0.51	0.38	0.05	MEANWIND	-0.08	0.63	-0.54	0.45	-0.03
MINSEA	-0.32	0.66	0.55	0.30	0.03	MAXSEA	-0.80	-0.14	0.29	0.42	0.09
MAXWIND	0.00	0.22	-0.20	-0.17	0.89	MEANSEA	-0.75	-0.27	0.34	0.44	0.08
MEANWIND	0.13	0.76	-0.31	-0.17	0.23	MINSEA	-0.68	-0.37	0.38	0.43	0.09
MINVIS	0.03	0.78	-0.29	-0.16	-0.02	PRE	0.13	0.49	0.10	-0.15	0.84
MAXVIS	-0.08	-0.34	-0.72	0.46	0.03						
MEANVIS	-0.19	-0.35	-0.71	0.41	-0.01						
PRECI	0.18	-0.46	-0.05	0.41	0.35						

As shown in Table 4, there was a strong positive correlation between the first component and Max-temperature (°C), Mean-temperature (°C), Min-temperature (°C), Max-dew point (°C), Mean-dew point (°C), Min-dew point (°C) in both areas. These indexes reflected the temperature of study areas. However, in Pearl River, there was a negative correlation between the first component and sea level pressure indexes while in Muskegon river it was positive correlation. The second to the fifth component in Muskegon river are the combination of other indexes, while the second to the fifth component in Pearl river are humidity indexes, wind indexes, sea level pressure and precipitation indexes respectively. The similarities and differences between principal components in Muskegon river and Pearl river showed that PCA method has the ability of extracting climatic characteristics in the different area. The result of PCA method is as Fig.4

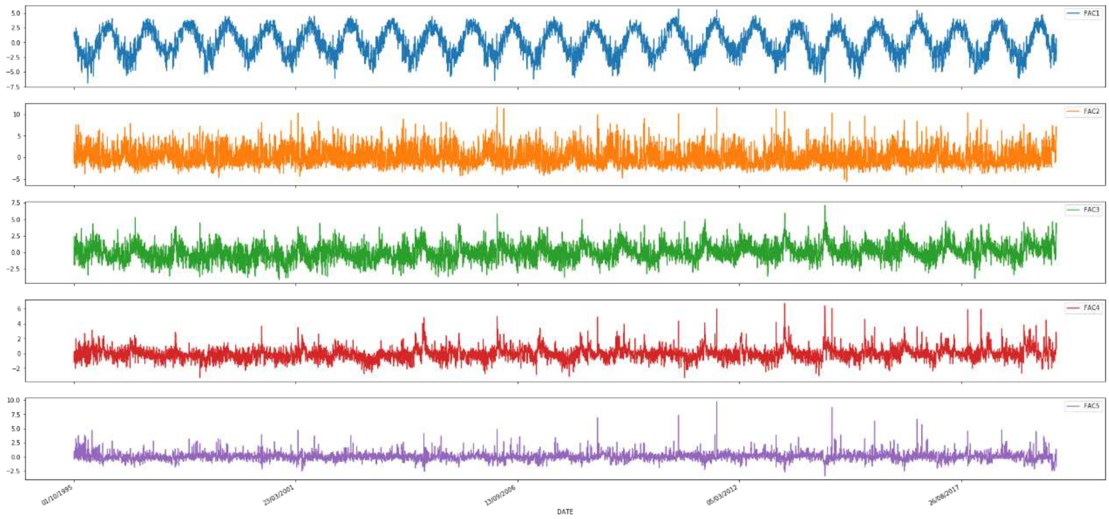

Figure 4a Five Principle components in Muskegon river

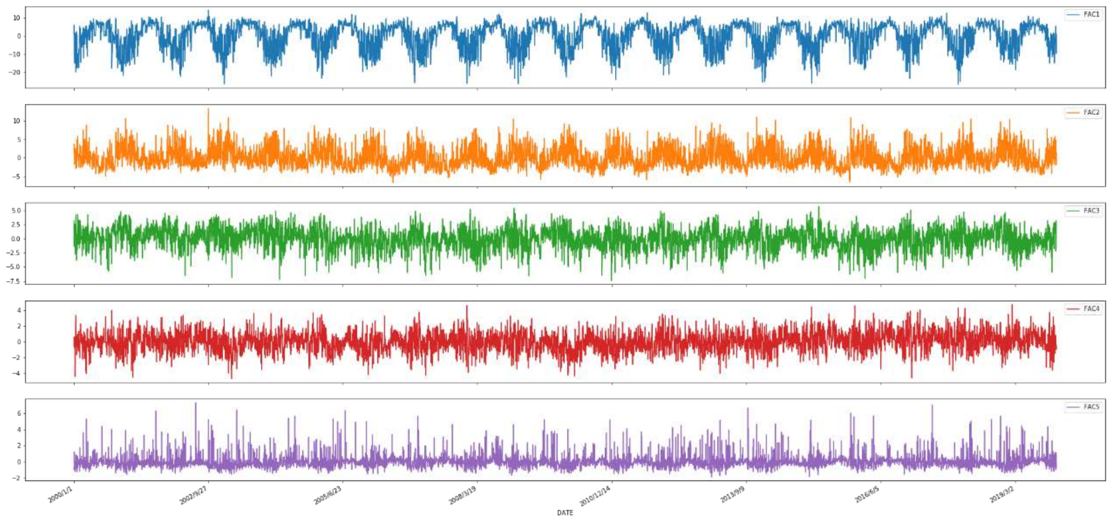

Figure 4b Five Principle components in Pearl river

3.2 Training for the ANN

Different RNN models (LSTM and GRU) were developed to test the influence of different input on runoff forecasting. These models were implemented using Python 3.7

and TensorFlow 2.0. The structure includes the input layer, hidden layer 1 with 16 hidden neurons, hidden layer 2 with 8 hidden neurons and output layer,. Dropout method was used between the hidden layers to deal with over fitting. Meanwhile, the RMSprop algorithm was used for training of LSTM and GRU in this study. The hyperparameter, past history, was set to 30 days, which means input data of every 30 days was used to predicate runoff of next day.

3.3 Performance of ANN

Different hidden block and input were compared to evaluate their effect on model performance and to identify the best-hidden block- input combination. 6 different scenarios were proposed as in Table 5 to predict the runoff. Ridge regression was used as a baseline model. These models were run on a computer with intel core i7-9750H CPU, 16GB memory.

Table 5 Different scenarios of hidden block and input

	Input	Hidden block kind
Scenario 1	Rainfall	LSTM
Scenario 2	Rainfall	GRU
Scenario 3	Multiple meteorological data	LSTM
Scenario 4	Multiple meteorological data	GRU

Scenario 5	Multiple meteorological data with PCA method	LSTM
Scenario 6	Multiple meteorological data with PCA method	GRU

Parts of the forecasting results were provided and compared with the baseline model in Figure 5a-d. Other results can be found in the support information (Figures S1-14). The model evaluation criteria results were provided in Table 6.

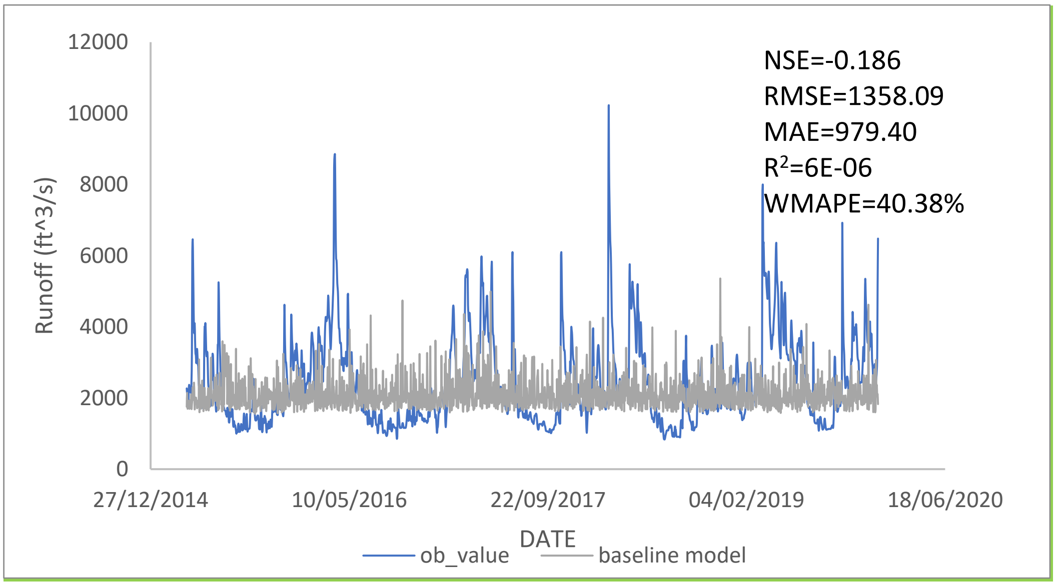

Figure 5a Baseline model: Ridge regression based on rainfall data in Muskegon river

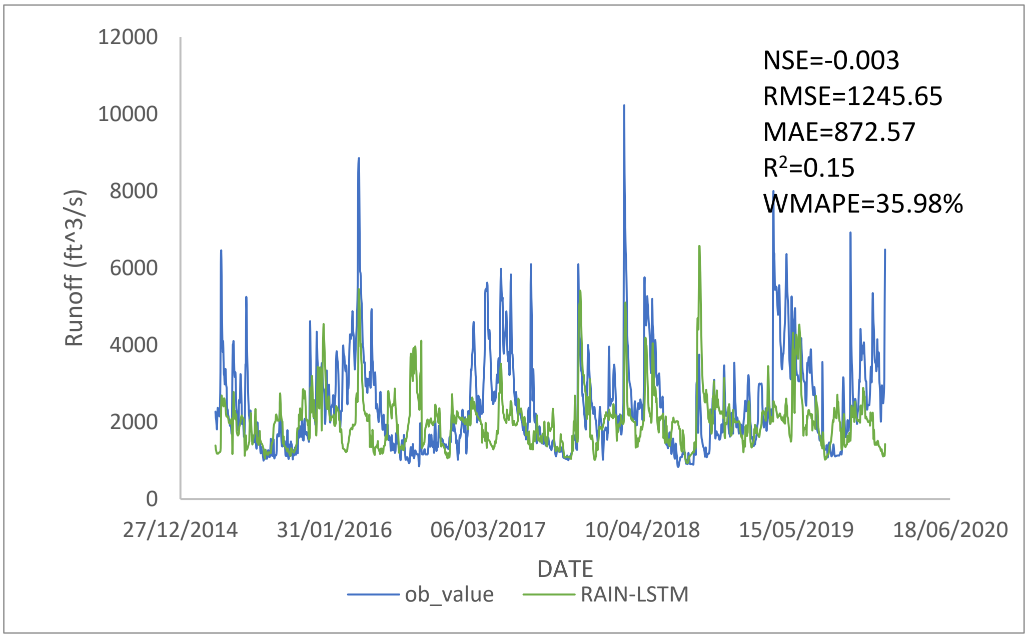

Figure 5b Scenario 1: LSTM neural network based on rainfall data in Muskegon river

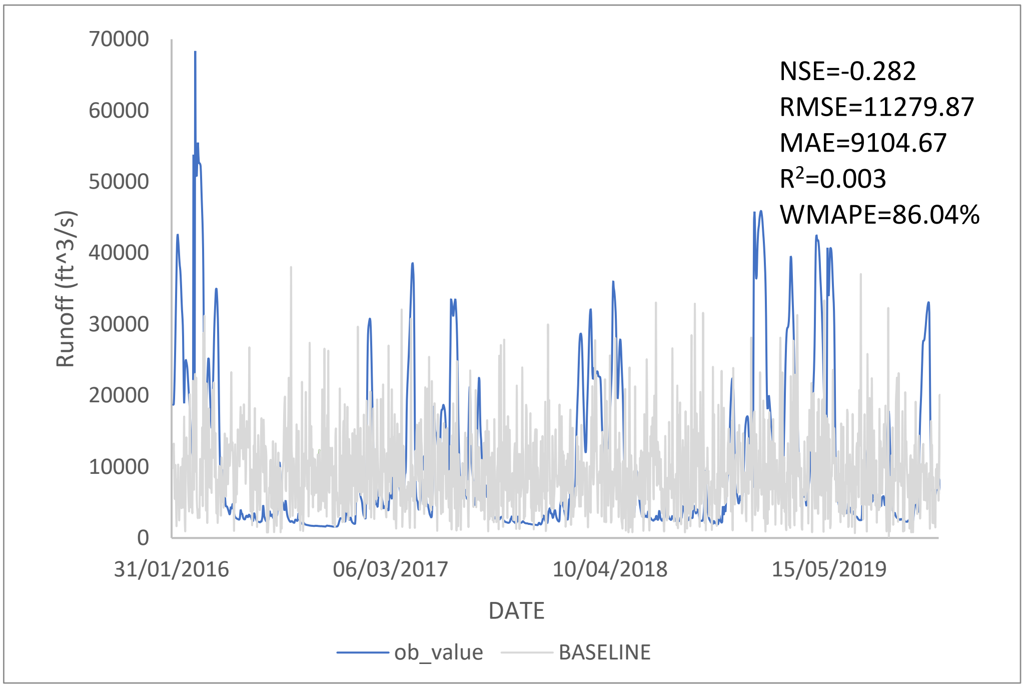

Figure 5c. Baseline model: Ridge regression based on rainfall data in Pearl river

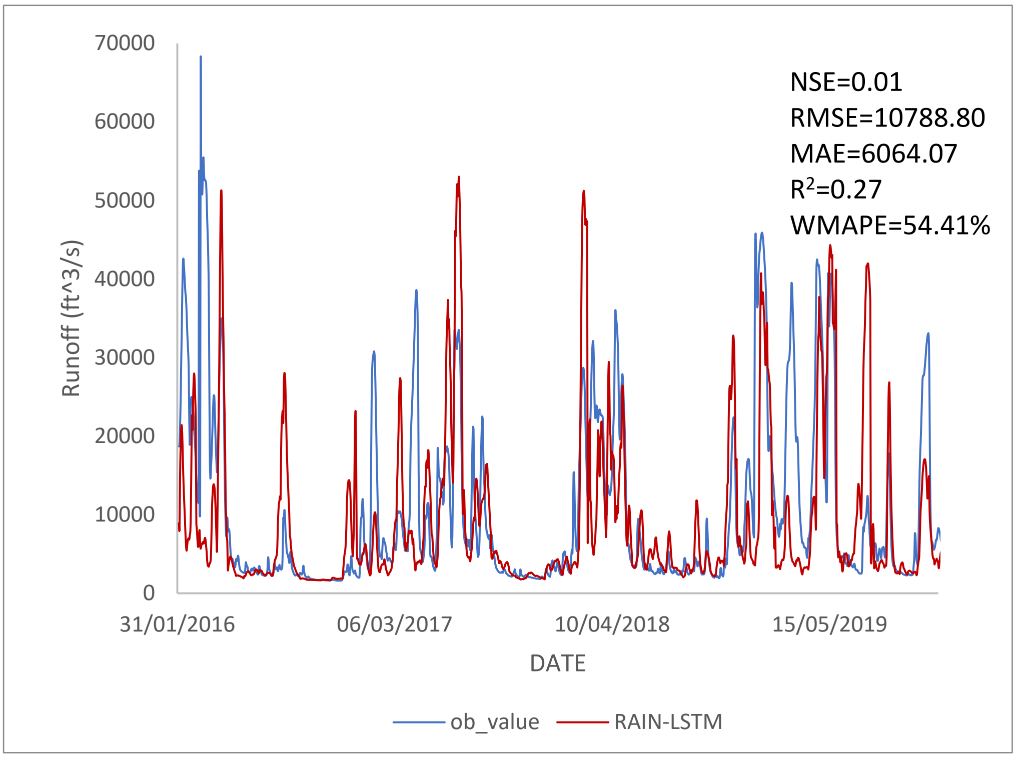

Figure 5d. Scenario 1: LSTM neural network based on rainfall data in Pearl river

Figure 5 shows runoff forecasting by different model and input. The observation value (blue line) is castrated with different models. (a) Ridge regression based on rainfall data in Muskegon river; (b)LSTM neural network based on rainfall data in Muskegon river; (c) Ridge regression based on rainfall data in Pearl river; (d) LSTM neural network based on rainfall data in Pearl river;

Table 6 Model evaluation result

	Baseline model	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6

	Input	Rainfall	Rainfall	Rainfall	multiple meteorological data	multiple meteorological data	multiple meteorological data with PCA method	multiple meteorological data with PCA method
	Hidden block kind	-	LSTM	GRU	LSTM	GRU	LSTM	GRU
Muskegon river	NSE	-0.186	0.003	-0.012	0.343	0.372	0.844	0.842
	RMSE(ft^3/s)	1358.09	1245.65	1254.90	1010.72	988.31	492.23	496.54
	MAE(ft^3/s)	979.40	872.57	895.68	674.75	659.16	276.65	279.60
	R^2	6E-06	0.15	0.127	0.46	0.49	0.85	0.84
	WMAPE	40.38%	35.98%	36.93%	27.18%	27.82%	11.41%	11.53%
	Time(s)	-	403.00	403.000	451.00	405.00	436.00	407.00
Pearl river	NSE	-0.2821	0.101	0.102	0.163	0.156	0.292	0.31
	RMSE(ft^3/s)	11279.87	10788.80	10781.84	10410.34	10454.19	8672.03	8549.07
	MAE(ft^3/s)	9104.67	6064.07	6164.57	6252.14	6341.64	5376.65	5107.27
	R^2	0.003	0.27	0.25	0.37	0.31	0.38	0.41
	WMAPE	86.04%	54.41%	57.90%	57.57%	58.44%	53.77%	55.06%
	Time(s)	-	410.00	407.000	442.00	409.00	435.00	412.00

As shown in Fig 5 and Table 6,all deep learning models have better performance than the baseline model with higher NSE and R^2 and lower RMSE, MAE and WMAPE, which prove the advantage and effectiveness of the deep learning model.

In Table 6, different input has a great influence on model accuracy. Deep RNN models with multiple meteorological data inputs (Scenario 3 and Scenario 4) had a better performance than rainfall data input only (Scenario 1 and Scenario 2). In both areas, NSEs of Scenario 1 and Scenario 2 were nearly 0, which means the results of deep RNN models with rainfall data input can only reflect the overall trend of runoff. Compared with Scenario 1 and Scenario 2, NSE and R^2in both areas were much higher in Scenario 3 and Scenario 4, meanwhile, RMSE and MAE reduced by nearly 20% in Muskegon river. The improvement of models in Pearl River was relatively small. This may be because multiple meteorological data included not only rainfall data, but also wind speed, temperature and other meteorological indicators that directly or indirectly affect the runoff generation process. This means that meteorological data can provide more effective information to achieve higher accuracy.

With the same hidden block, the accuracy of deep RNN model with PCA input (Scenario 5 and Scenario 6) may outperform model with normal multiple meteorological data inputs (Scenario 3 and Scenario 4) NSE and R^2 of Scenario 5 and Scenario 6 were nearly twice as much as Scenario 3 and Scenario 4 in both areas. Meanwhile, in Muskegon river, RMSE, MAE and WMAPE were nearly 50% less than Scenario 3 and Scenario 4. This means PCA method can reflect core information by classifying the original data information into several comprehensive variables and prevent the interference of useless information.

With the same input, deep GRU model can achieve the same accuracy as deep LSTM model, and reduce the computational load. This phenomenon is more obvious when processing high-dimensional input data. When the input data were just rainfall (Scenario 1 and Scenario 2), the calculation time of deep LSTM model and deep GRU model were the same. With the input data changed to PCA data (Scenario 5 and Scenario 6) and multiple meteorological data (Scenario 3 and Scenario 4), the calculation time of deep LSTM model rose dramatically to 436s and 451s in Muskegon River and 435s and 442s in Pearl River, while the calculation time of deep GRU model ascended slightly to 405s and 407sin Muskegon River and 407s and 412s in Pearl River. This phenomenon could be due to the structure of the hidden block. The number of parameters which need to be identified in each GRU block is 9 (6 weights and 3 biases) while 12 parameters (8 weights and 4 biases) in each LSTM block need to be trained. With the same optimization method and input data, the less the number of identification parameters, the faster to get the optimal solution.

4. Conclusion

In order to evaluate the influence of different input variable on runoff forecasting by RNN approach and identify the best input, Muskegon River and Pearl River were taken as examples. Rainfall data, multiple meteorological data and multiple meteorological data with PCA method were considered as input of deep LSTM model and deep GRU model. Four evaluation criteria were employed to evaluate the influence of different input variables on the accuracy of model quantitatively.

Several key conclusions can be made as follows:

1) The selection of model inputs has a great influence on model accuracy. Deep RNN model with multiple meteorological data inputs achieves higher accuracy than rainfall data input for runoff forecasting.
2) PCA method can be applied to improve the accuracy of deep RNN model effectively as it can reflect core information by classifying the original data information into several comprehensive variables.
3) The accuracy of deep LSTMs model and deep GRUs model is much the same, but the computational load of deep GRUs model is lower, especially with high dimension input.

Although ANNs or DNNs have a astonishing performance in hydroloy and we get satisfactory result, the black-box nature of ANNs or DNNs is still a barrier to application. In this study, the rainfall-runoff process between meteorological data and rainfall was fitted by LSTMs model and deep GRUs model and the noise in input data was filtered by PCA method. We believe that the surprising accuracy of model performance and increasing studies on ANNs would increase the trust in data-driven approaches and lead to more practices in hydrologic sciences. Meanwhile, it is still a significant work to consider the constraints of physical process in data-driven model and explain the physical meaning of parameters and hyperparameters in the data-driven model.

Acknowledgement

This research was funded by the National Natural Science Foundation of China (Grant No. 91747204) and NERC case studentship. Datasets for this research are available in these in-text data citation references: runoff (USGS 2020) and meteorological data (WeatherUnderground 2020). We thank the Associate Editor and two anonymous reviewers for the constructive comments. The manuscript has been significantly improved by incorporating these suggestions.

References

Amiri, E., 2015: Forecasting daily river flows using nonlinear time series models. *Journal of Hydrology*, **527,** 1054-1072.

Bittelli, M., F. Tomei, A. Pistocchi, M. Flury, J. Boll, E. S. Brooks, and G. Antolini, 2010: Development and testing of a physically based, three-dimensional model of surface and subsurface hydrology. *Advances in Water Resources*, **33,** 106-122.

Chen, B.-W., N. N. B. Abdullah, S. Park, and Y. Gu, 2018: Efficient multiple incremental computation for Kernel Ridge Regression with Bayesian uncertainty modeling. *Future Generation Computer Systems-the International Journal of Escience*, **82,** 679-688.

Chen, X., and Coauthors, 2020: The importance of short lag-time in the runoff forecasting model based on long short-term memory. *Journal of Hydrology*, **589**.

Cheng, M., F. Fang, T. Kinouchi, I. M. Navon, and C. C. Pain, 2020: Long lead-time daily and monthly streamflow forecasting using machine learning methods. *Journal of Hydrology*, **590**.

Cho, K., B. Van Merriënboer, D. Bahdanau, and Y. J. a. p. a. Bengio, 2014: On the properties of neural machine translation: Encoder-decoder approaches.

Chung, J., C. Gulcehre, K. Cho, and Y. J. a. p. a. Bengio, 2014: Empirical evaluation of gated recurrent neural networks on sequence modeling.

de la Fuente, A., V. Meruane, and C. Meruane, 2019: Hydrological Early Warning System Based on a Deep Learning Runoff Model Coupled with a Meteorological Forecast. *Water*, **11**.

Hochreiter, S., and J. Schmidhuber, 1997: Long short-term memory. *Neural Computation*, **9,** 1735-1780.

Hoerl, A. E., 1959: Optimum solution of many variables equations. *Chemical Engineering Progress*, **55,** 69-78.

Hotelling, H., 1933: Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24,** 498-520.

Hu, C. H., Q. Wu, H. Li, S. Q. Jian, N. Li, and Z. Z. Lou, 2018: Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *Water*, **10**.

Hyndman, R. J., and G. Athanasopoulos, 2018: *Forecasting: principles and practice.* OTexts.

Jeong, J., and E. Park, 2019: Comparative applications of data-driven models representing water table fluctuations. *Journal of Hydrology*, **572,** 261-273.

Jeong, J., and Coauthors, 2017: A predictive estimation method for carbon dioxide transport by data-driven modeling with a physically-based data model. *Journal of Contaminant Hydrology*, **206,** 34-42.

Ji, S. W., W. Xu, M. Yang, and K. Yu, 2013: 3D Convolutional Neural Networks for Human Action Recognition. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, **35,** 221-231.

Johengen, T. H., B. A. Biddanda, and J. B. Cotner, 2008: Stimulation of Lake Michigan plankton metabolism by sediment resuspension and river runoff. *Journal of Great Lakes Research*, **34,** 213-227.

Kao, I. F., Y. Zhou, L.-C. Chang, and F.-J. Chang, 2020: Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, **583**.

Le, X. H., H. V. Ho, G. Lee, and S. Jung, 2019: Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water*, **11**.

Li, X. L., H. S. Lu, R. Horton, T. Q. An, and Z. B. Yu, 2014: Real-time flood forecast using the coupling support vector machine and data assimilation method. *Journal of Hydroinformatics*, **16,** 973-988.

Litjens, G., and Coauthors, 2017: A survey on deep learning in medical image analysis. *Medical Image Analysis*, **42,** 60-88.

Magassouba, A., K. Sugiura, and H. Kawai, 2020: A Multimodal Target-Source Classifier With Attention Branches to Understand Ambiguous Instructions for Fetching Daily Objects. *Ieee Robotics and Automation Letters*, **5,** 532-539.

Mehdizadeh, S., F. Fathian, and J. F. Adamowski, 2019: Hybrid artificial intelligence-time series models for monthly streamflow modeling. *Applied Soft Computing*, **80,** 873-887.

Miche, M., E. Studerus, A. H. Meyer, A. T. Gloster, K. Beesdo-Baum, H.-U. Wittchen, and R. Lieb, 2020: Prospective prediction of suicide attempts in community adolescents and young adults, using regression methods and machine learning. *Journal of Affective Disorders*, **265,** 570-578.

Napolitano, G., F. Serinaldi, and L. See, 2011: Impact of EMD decomposition and random initialisation of weights in ANN hindcasting of daily stream flow series: An empirical examination. *Journal of Hydrology*, **406,** 199-214.

Partington, D., P. Brunner, C. T. Simmons, A. D. Werner, R. Therrien, H. R. Maier, and G. C. Dandy, 2012: Evaluation of outputs from automated baseflow separation methods against simulated baseflow from a physically based, surface water-groundwater flow model. *Journal of Hydrology*, **458,** 28-39.

Pulukuri, S., V. R. Keesara, and P. Deva, 2018: Flow Forecasting in a Watershed using Autoregressive Updating Model. *Water Resources Management*, **32,** 2701-2716.

Raghu, M., B. Poole, J. Kleinberg, S. Ganguli, and J. S. Dickstein, 2017: On the expressive power of deep neural networks. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2847-2854.

Ray, D. K., J. M. Duckles, and B. C. Pijanowski, 2010: The impact of future land use scenarios on runoff volumes in the Muskegon River Watershed. *Environmental management*, **46,** 351-366.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, 2019: Deep learning and process understanding for data-driven Earth system science. *Nature*, **566,** 195-204.

Riad, S., J. Mania, L. Bouchaou, and Y. Najjar, 2004: Rainfall-runoff model using an artificial neural network approach. *Mathematical and Computer Modelling*, **40,** 839-846.

Solomatine, D. P., and A. Ostfeld, 2008: Data-driven modelling: some past experiences and new approaches. *Journal of hydroinformatics*, **10,** 3-22.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15,** 1929-1958.

Taylor, K. L., and J. B. Grace, 1995: THE EFFECTS OF VERTEBRATE HERBIVORY ON PLANT COMMUNITY STRUCTURE IN THE COASTAL MARSHES OF THE PEARL RIVER, LOUISIANA, USA. *Wetlands*, **15,** 68-73.

USGS, cited 2020: National Water Information System [Available online at https://maps.waterdata.usgs.gov.]

Valipour, M., 2015: Long-term runoff study using SARIMA and ARIMA models in the United States. *Meteorological Applications*, **22,** 592-598.

Wang, Q., and Coauthors, 2020: Sequence-based statistical downscaling and its application to hydrologic simulations based on machine learning and big data. *Journal of Hydrology*, **586**.

Wang, W. C., K. W. Chau, C. T. Cheng, and L. Qiu, 2009: A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of Hydrology*, **374,** 294-306.

WeatherUnderground, cited 2020: Weather Underground Historical Weather. [Available online at https://www.wunderground.com/.]

Yoon, H., S. C. Jun, Y. Hyun, G. O. Bae, and K. K. Lee, 2011: A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. *Journal of Hydrology*, **396,** 128-138.

Yuan, X., C. Chen, X. Lei, Y. Yuan, and R. Muhammad Adnan, 2018: Monthly runoff forecasting based on LSTM–ALO model. *Stochastic Environmental Research and Risk Assessment*, **32,** 2199-2212.

Zhongrun, X., and D. Ibrahim, 2020: Distributed long-term hourly streamflow predictions using deep learning - A case study for State of Iowa. *Environmental Modelling & Software*, **131**.

Zhu, S., J. Z. Zhou, L. Ye, and C. Q. Meng, 2016: Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze River, China. *Environmental Earth Sciences*, **75**.

542