Modeling Earth Systems and Environment (2020) 6:2153–2164
https://doi.org/10.1007/s40808-020-00830-w

ORIGINAL ARTICLE

Impact of training data size on the LSTM performances for rainfall–runoff modeling

T. Boulmaiz[1] · M. Guermoui[2] · H. Boutaghane[3]

Received: 11 March 2020 / Accepted: 24 May 2020 / Published online: 2 June 2020
© Springer Nature Switzerland AG 2020

Abstract

For predicting catchment runoff with data-driven methods, a long historical database of measurements is required. The current study focuses on the assessment of a deep learning model named long short-term memory (LSTM) for rainfall–runoff relationship with different training data size. The developed model has been evaluated on twenty catchments with diverse hydrological conditions obtained from the freely available CAMELS dataset. In order to prove the efficiency of the proposed model for runoff prediction, we test its performances against the traditional feed-forward neural network model. The studied models have been trained with the same input parameters and different size of training data to show the effect of data length on the prediction performances. To this end, the length of training data was varied from 3 to 15 years, while the model was tested on 10 years of data. The results show that the deep LSTM outperforms the traditional model in terms of statistical indicators over different size of training sets. The proposed deep LSTM model can predict runoff with acceptable performances using 9 years of data length in the training procedure, a result that improves when using 12 years. In addition, it has been proven that the deep LSTM model may be efficient even when using small data size (3 years) compared to its benchmarked model which require 9 years for similar results. Thus, the LSTM network is a powerful deep learning model able to learn the behavior of rainfall–runoff relationship with a minimum data length.

Keywords CAMELS · Data length · Deep learning · FFNN · Long short-term memory · Rainfall–runoff modeling

Introduction

Hydrological modeling requires a high accuracy in estimating runoff for effective water resources planning and flood protection. Data-driven machine learning approaches have experienced a rapidly development in the last 20 years. These methods have been widely used for their reliability in rainfall–runoff modeling (Remesan and Mathew 2015). These models are based mainly on a learning approach to simulate catchment behavior of generating runoff. However, this operation is a challenging task especially for regions with limited measurements. The main constraint encountered in modeling rainfall–runoff in these regions is highly linked to the degraded quality of the data which stems from unreliable measurements that can manifest in insufficient historical data length or the presence of an important rate of gaps.

Data sequences which containing more hydrological variability are preferred to enhance the performance of the calibrated model (Gupta and Sorooshian 1985a, b). Logically, the improvement of model performance depends on the increasing of data length since it is more suitable to contain a huge amount of information. Several studies (Brath et al. 2004; Merz et al. 2009; Perrin et al. 2007) on model calibration agree that there is a minimum required length to develop a robust model; however, beyond a certain limit, the improvement is not significant. It has been showed that a parsimonious model calibration may yield acceptable results using less than 1 year of data (Brath et al. 2004; Perrin et al. 2007). The effects of timescale in hydrological modeling study (Merz et al. 2009) have been concluded by a suggestion of a calibration period of 5 years to capture most of

✉ T. Boulmaiz
boulmaiz.tayeb@univ-ghardaia.dz; t.boulmaiz@gmail.com

[1] Materials, Energy Systems Technology and Environment Laboratory, Ghardaia University, Ghardaia, Algeria

[2] Unité de Recherche Appliquée en Energies Renouvelables, URAER, Centre de Développement des Energies Renouvelables, CDER, 47133 Ghardaïa, Algeria

[3] Hydraulic Department, Badji Mokhtar-Annaba University, P. O. Box 12, Annaba, Algeria

the temporal hydrological variability. The study also settled an upper limit of 15 years as they observed no significant improvement by using a wider calibration period. Nevertheless, with the use of data-driven model such as the feed-forward neural network (FFNN) showed that the performance of the model kept on improving while increasing the length of the data sample (Anctil et al. 2004).

Out of all the learning techniques and architectures inspired from the human brain in its simplified version, neural network has been the most suitable method based on soft computing techniques for the estimation of complex rainfall–runoff behavior (Chen et al. 2013; Jeong and Kim 2005; Solaimani 2009). A study based on the use FFNN for daily runoff modeling revealed that this model provides low modeling error compared with regression and conceptual models (Tokar and Johnson 1999). Although conventional neural networks model shows high prediction accuracy, the precision reached is still unsatisfactory when facing a highly non-stationary behavior of time series. This constraint has been the focus of several researches conducted in the last decade until the introduction of the long short-term memory (LSTM) model. This latter, which has been developed since 1997 (Hochreiter and Schmidhuber 1997), was gradually emerged with the advance of technology. The LSTM neural network model differs from the conventional neural network by introducing the memory cells and the memories of the previous input in the forecasting process. This type of deep neural network has proven its utility as it was applied in many fields such as: time series prediction (Wang et al. 2018), emotion analysis (Wöllmer et al. 2013) and natural language processing (Graves et al. 2013). The results of these works proved the ability of LSTM learning of nonlinear time series to be one of the most powerful models for forecasting problems.

The principal purpose behind this study is to prove the performance of LSTM model for modeling rainfall–runoff relationship. Out of the studies presented in the literature, there are few works using LSTM model in the field of modeling rainfall–runoff. Compared to a conceptual model (Sacramento Soil Moisture Accounting Model), it has been shown similar performances by using the LSTM model on daily rainfall runoff simulation, either for individual catchment or as a regional hydrological model (Kratzert et al. 2018). Another hydrological application using the LSTM was carried out at hourly scale (Hu et al. 2018), which demonstrated that the LSTM outperformed the FFNN during different flooding events on a river located in northern China. The same conclusion has been obtained by comparing this deep learning method with GR4H model in predicting hourly runoff (Ayzel 2019).

In this study, an assessment of the LSTM model on capturing the rainfall runoff relationship is carried out using the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset, which was obtained freely from the website of the National Center of Atmospheric Research (NCAR) (https://ral.ucar.edu/solutions/products/camels). This work distinguishes itself by considering the learning data size impact on the LSTM performances. An implementation of a FFNN model by using the same LSTM inputs served as a benchmark model to validate the performance of the developed model against conventional machine learning techniques.

Datasets

Twenty forest catchments (Table 1) with different areas and slopes have been selected from the CAMEL dataset in order to test the performance of the studied LSTM model. Catchment Attributes for Large-Sample Studies CAMELS dataset is a research community database of daily forcing (precipitation, solar radiation, minimum/maximum temperature and humidity) and hydrological response data for 671 small- to medium-sized basins across the contiguous USA. These catchments are less disturbed by human activities (Addor et al. 2018; Newman et al. 2015). The CAMELS dataset describes six main classes of attributes at the catchment scale: topography, climate, streamflow, land cover, soil and geology (Newman et al. 2015).

The dataset provides meteorological data which cover the period between (1980 and 2014). The daily forcing data are obtained from the Daymet gridded data source (Thornton et al. 2012). This product generates gridded estimates (1 km x 1 km) of daily weather parameters derived from daily meteorological observations by interpolation and extrapolation procedures.

The annual meteorological behavior in the catchments used in this study is summarized in (Fig. 1). As shown in the figure, catchments with different meteorological conditions are selected for the study purpose.

Theoretical overviews

In this section we provide a short description of the studied models (LSTM, FFNN). The theoretical explanations of LSTM and FFNN are all explained elsewhere since they are well-known techniques (see Aggarwal 2018 for detailed descriptions).

Feed-forward neural network (FFNN)

The FFNN model is the most used model for data prediction. It consists of three principal layers (see Fig. 2): (1) input layer, (2) hidden layer and (3) output layer. Each product of

Table 1 Catchments physical characteristics and gauging stations information

Catchment no.	Area (km^2)	Slope (%)	Fraction forest (%)	Gauging station							
				Lat.	Long.	Min (m^3/s)	Mean (m^3/s)	Max (m^3/s)	Std	Cv	Sk
1	2304	22	91	47.24	−68.58	1.2	43	507	52	1.2	2.6
2	620	18	92	44.61	−67.94	0.3	14	192	18	1.3	3.4
3	3676	13	88	45.50	−68.31	0.9	76	733	103	1.4	2.5
4	767	30	96	45.18	−69.31	0.2	18	898	30	1.7	6.2
5	905	50	99	44.87	−69.96	0.8	23	1019	37	1.6	5.7
6	396	60	100	44.88	−71.06	0.3	10	280	17	1.7	4.8
7	216	33	94	44.30	−70.54	0	4	191	6.7	1.6	6.5
8	38	5	90	43.15	−70.97	0	0.6	27	1.1	1.9	8.8
9	347	50	99	43.57	−71.75	0.1	4.4	111	6.5	1.5	4.5
10	451	48	100	44.51	−71.84	0.1	4.6	116	6.8	1.5	5
11	785	110	100	44.27	−71.63	0.5	6.3	137	8.9	1.4	5
12	373	43	97	44.15	−72.07	0.2	4.7	68	5.7	1.2	3.7
13	518	41	100	44.09	−72.34	0	0.5	7.6	0.5	1.1	3.3
14	347	47	98	43.93	−72.66	0	1.6	44	2.1	1.3	4.6
15	460	60	98	43.71	−72.42	1.9	38	937	50	1.3	4.4
16	310	19	97	42.68	−72.12	0	1	22	1.4	1.3	3.5
17	375	50	100	42.64	−72.73	0.2	5.9	270	10	1.7	7.6
18	326	49	100	42.70	−72.67	0.1	2.6	95	3.8	1.4	6
19	418	33	99	42.24	−72.90	0.1	5.9	405	11	2	12
20	391	41	100	42.71	−73.20	0.4	6.2	91	8.3	1.3	3.6

Lat., Long., Std, Cv and Sk are, respectively, latitude, longitude, standard deviation, coefficient of variation and skewness coefficient

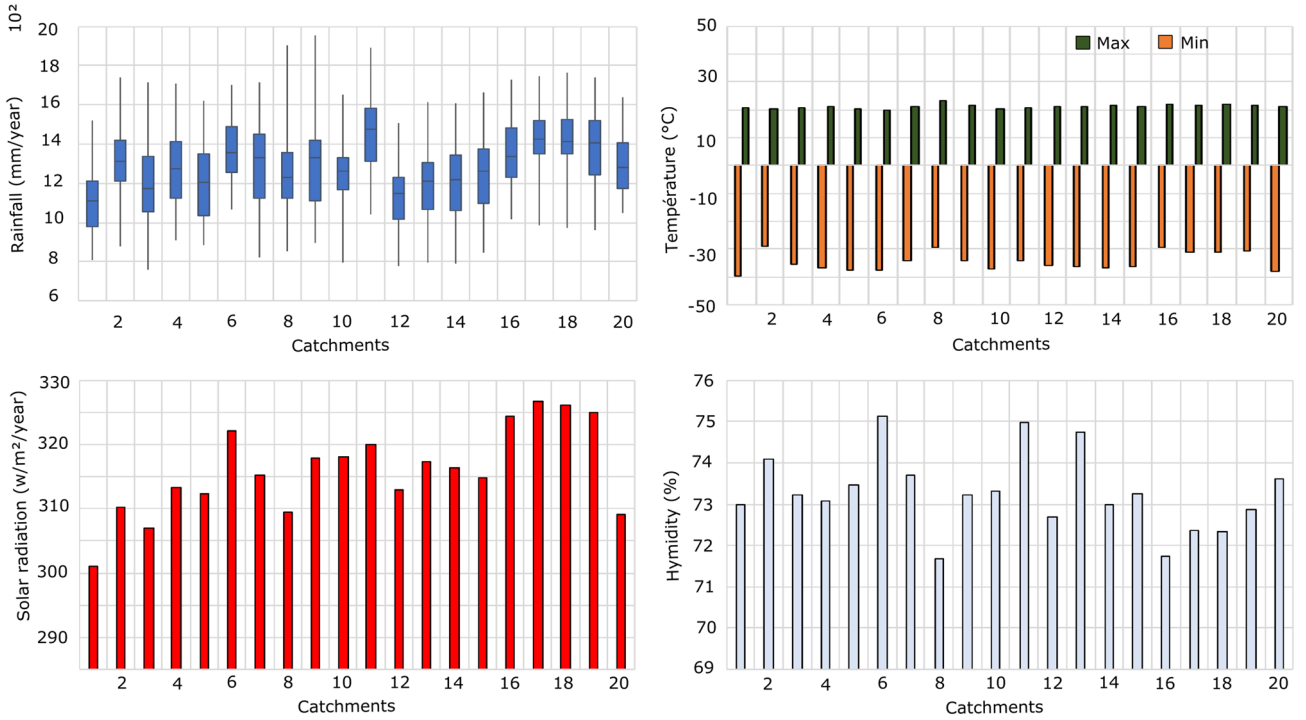

Fig. 1 Annual meteorological variation in the catchments used in this study

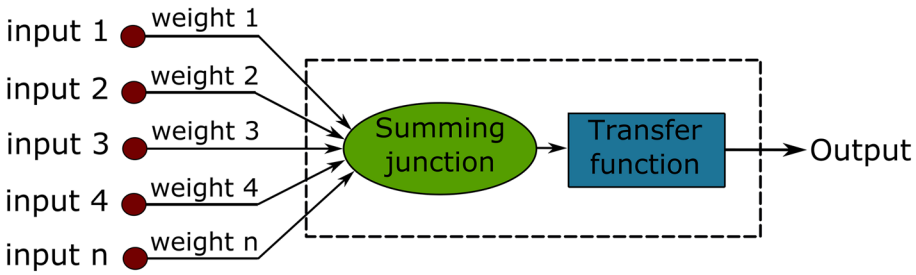

Fig. 2 Basic structure of FFNN (Ghimire et al. 2019)

input data (a_i) and weights (w_{ij}) is summed with the bias (b_j) according to Eq. (1):

$$x = \left(\sum_{i=1}^{n} w_{ij} a_i\right) + b_i \tag{1}$$

Transfer function (F) in hidden layer (tansig or logsig) is then applied, to generate the desired output Eq. (2).

$$F(x) = F\left[\left(\sum_{i=1}^{n} w_{ij} a_i\right) + b_j\right] \tag{2}$$

In the training process, the algorithm adjusts the weights and biases iteratively to boost the estimation performance of the outputs (Ghimire et al. 2019). The training period is repeated until the error between estimated and measured value reaches pre-defined thresholds (Guermoui et al. 2016). Before the training procedure, the number of neurons in the hidden layer and the input sequence length (delays) have to be fixed. Many learning algorithms are proposed in the literature. The most used one is Levenberg–Marquardt (Levenberg 1944; Marquardt 1963).

Long short-term memory (LSTM)

The LSTM model is based mainly on the use of dependence between consecutive events on a relevant time step (Ghimire et al. 2019). The LSTM neural network model is a part of deep recurrent neural network (RNN) as shown in Fig. 3.

In the RNN mechanism, the hidden units share information based on a time index. The sharing process helps in building memory blocks of long time series which help the model to recognize and predict the sequences. From (Fig. 3), the feedback loop provides the units with memory, in which the previous states of the current neuron can be used as input parameters when updating memory. LSTM model contains three main units which are defined as input, output and forget gates. The units construct the memory blocks to provide an ability to update and filter information flow in different blocks (Chen et al. 2018). The operation system of LSTMs model can be summarized as follow. Firstly, from the new input data x_t and the last hidden state h_{t-1}, the model is able to select the information to be deleted from the cell state which is represented by the forget gate f_t according to Eq. (3):

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{3}$$

In Eq. (3), W_f represents the weigh matrices;b_f is the bias vector; and $\sigma(\ldots)$ is the logistic sigmoid function.

In the second step, the models decide which information must be memorized in the cell state. To this end, a new candidate cell state $\tilde{C}_t$ is performed and scaled by the input gate.

$$\tilde{C}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{4}$$

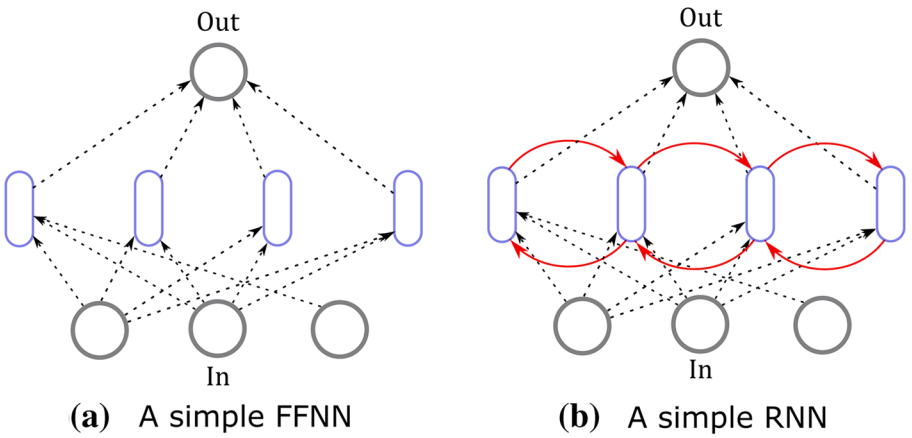

Fig. 3 Simple FFNN model versus RNN model (Srivastava and Lessmann 2018)

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \quad (5)$$

In the following step, the new cell state C_t is updated by combining the new cell state $\tilde{C}_t$ and the previous cell state C_{t-1}, where the previous is effected by forget gate and the new state is scaled by input gate i_t:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

The last step is divided into two different actions. A new gate called output gate O_t is applied to select the suitable part of cell state to be outputted. The cell state C_t is activated by the tanh function, the multiplication results is the desired output h_t:

$$O_t = \sigma\left(W_0 \cdot \left[h_{t-1}, x_t\right] + b_0\right) \quad (7)$$

$$h_t = O_t \cdot \tanh\left(C_t\right) \quad (8)$$

As with classical neural network where some parameters called "hyperparameter" have to be fixed before the training procedure, the LSTM network has the number of layer, the hidden units and the input sequence (batch size). Usually, one or two LSTM layers are stacked with a defined number of hidden units. These two hyperparameters are related to the studied application complexity. In the field of rainfall runoff modeling, Kratzert et al. (2018) used two layers and 20 hidden units at each layer while (Hu et al. 2018) performed a hourly prediction using one layer and 50 hidden units. For the batch size, the first opted for 365 days in the aim to capture at least the dynamics of a full annual cycle. In the second study, Hu et al. (2018) proceeded to the try and error approach in the range from 1 to 6 h. In addition of the LSTM layer, a dense layer (fully connected layer) is stacked to map the LSTM layer output to a desired output size. Due to the complexity of deep learning models, a dropout layer may be added to the network to relieve the possible overtraining problem. This achievement is carried out by excluding a rate of LSTM units from activation and weight updates during the training process (Fig. 4).

In order to train the LSTM model, the recently developed stochastic optimization algorithm "Adam" (Kingma and Ba 2014) has been used. This latter gained popularity in the field of deep learning due to its efficiency and rapidity. Adam algorithm has been recommended by Ruder (2016) who developed a comprehensive review of modern gradient descent optimization algorithms.

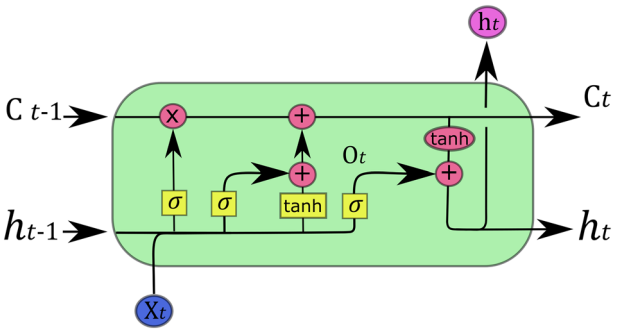

Fig. 4 LSTM neural network model

General methodology

For an unbiased comparison, the datasets are divided in the same way for each developed model. A dataset of the last 10 years which represent 33% from the whole data is selected to test the LSTM and FFNN models. By using a long period for testing the models, we expect the existence of a wide variety of small, medium and large events in these datasets which may avoid a biased assessment of their performances. Five years (17%) preceding the test period are used as a validation procedure to prevent overfitting. The problem of overfitting occurs when training performances are maximized by creating a complex model which overfit the learning data. To prevent this situation, instead of using a training/test split, we will use a training/validation/test split. The aim of this procedure (also known by the holdout method) is to compute the error estimation of the network at each iteration of an unseen data (validation data) and stop the training process when the error rate of validation data is increasing. Finally, the learning phase is carried out with four variants in order to investigate the data length impact on the efficiency of models. The five variants evaluated are created from the first 15 years (50%) of datasets, beginning by the last 3 years and extend at each variant by 3 years. The five different data division strategies are illustrated in (Fig. 5).

The performance of neural networks models is highly related to reaching an optimal structure able to map the process effectively. Some hyperparameter have to be fixed before the training procedure, this issue has been resolved in most case by the try-error approach which costs time but is more efficient to reach high accuracy. Table 2 shows more details about the configuration of both networks. Levenberg–Marquardt and Adam algorithms have been, respectively, used for training the FFNN and LSTM networks.

Efficiency criteria

In the aim of evaluating the performances of the LSTM and FFNN models, four widely used efficiency criteria are used: Nash–Sutcliff efficiency (NSE), root mean square error (RMSE), determination coefficient (R^2) and the bias (Eqs. 9, 10 and 11, respectively).

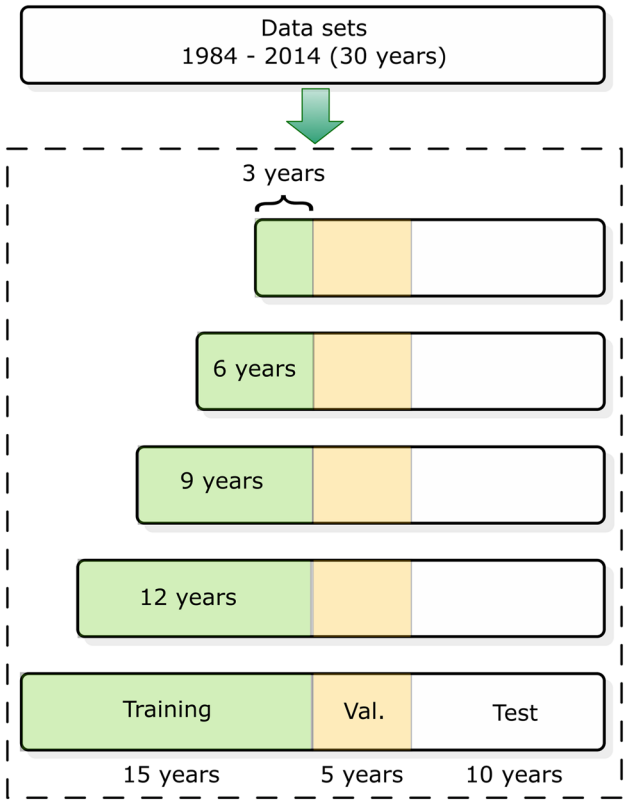

Fig. 5 Data split strategies used in this study

$$\mathrm{NSE} = 1 - \left[\frac{\sum_{i=1}^{n}\left(Q_{o_i} - Q_{p_i}\right)^2}{\sum_{i=1}^{n}\left(Q_{o_i} - \overline{Q_o}\right)^2}\right] \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Q_{o_i} - Q_{p_i}\right)^2} \tag{10}$$

$$R^2 = \frac{\left[\sum_{i=1}^{n}\left(Q_{o_i} - \overline{Q_o}\right)\cdot\left(Q_{p_i} - \overline{Q_p}\right)\right]^2}{\sum_{i=1}^{n}\left(Q_{o_i} - \overline{Q_o}\right)\cdot\sum_{i=1}^{n}\left(Q_{p_i} - \overline{Q_p}\right)} \tag{11}$$

where Q_{o_i}, Q_{p_i} and n are, respectively the observed, predicted streamflow and data number. $\overline{Q_o}$ and $\overline{Q_p}$ are the means of observed and predicted streamflow, respectively.

NSE (Nash–Sutcliff efficiency), which is the most used criteria by hydrologists, expresses the fraction of the variance of the runoff explained by the proposed model. The closer the NSE value is to 1, the better is the agreement between prediction and measurements. A negative value of the NSE indicates that it is better to use the mean observed runoff as the model rather than the model proposed. A value of 0 indicates that the proposed model performance is the same as the mean observed runoff. NSE criteria are sensitive to extreme values due to squared difference between predictions and measurements, larger values in a time series strongly affect the NSE criteria. The RMSE (root mean square error) is the measure of the differences between predicted and observed values in the unit of the variable (m^3/s in this study). A zero value of RMSE signifies that there is a perfect fit between the predicted and observed values. When it is increased, it indicates a lower performance of the model.

Table 2 Configuration of the FFNN and LSTM networks used in this study

Type of network	Hyperparameter	Configuration in this study
FFNN	Layers	1. Input layer 2. Hidden layer 3. Output layers
	Hidden units	Determined by try-error procedure in the range of [3–20]
	Delayed inputs	Determined by try-error procedure in the range of [1–5]
	Iteration epochs	Important to avoid the overfitting problem, however, by using the holdout method, the training iteration epoch is no longer relevant
LSTM	Layers	1. Input layer 2. Hidden LSTM layer: one LSTM layer was found to be sufficient after preceding tests 3. Dropout layer: with a rate of 20% 4. Dense layer 5. Output layer
	Hidden units	Determined by try-error procedure in the range of [10–50]
	Input sequence	Since the dynamic of catchments is varying from one to other, the input sequence involved to predict runoff at each time step has to be varied Determined by try-error procedure with values of 7, 30, 180 and 365 days

Finally, the R^2 (Coefficient of determination) describes the degree of the measured data variance explained by the proposed model. As example, the $R^2 = 0.5$ means that the model explains 50% of the variance in the observation data series.

Results and discussion

LSTM and FFNN models have been performed to runoff simulation by using rainfall and other climatic variables (minimal and maximal temperature, solar radiation and humidity) as inputs. Several model configurations (mini-batch and number of hidden units for the LSTM, delays and number of hidden neurons for the FFNN) have been selected to reach best performances.

An overview of the achieved performances after the learning procedure of LSTM and FFNN methods is graphically presented in (Fig. 6). The data length served for training both models has been varied from 3 to 15 years. An important difference is observed in terms of NSE statistics between the two studied models. The LSTM is robust against all cases of learning size and outperforms the benchmarked model. In going from 3 to 15 years, NSE statistics (1st quartiles, medians and 3rd quartiles) of the first model are significantly higher than those of the second model (e.g., medians of the LSTM are ranged in [0.61–072], while the FFNN is ranged in [0.55–0.64]). However, in terms of performance improvement, it can be observed that the increase in training data size is less effective from 9 years for the LSTM and almost insignificant from 12 years for both models. On the other hand, when increasing the training size, we can observe a change in the sample distribution of NSE values of the LSTM, especially from 6 to 9 years of learning data. The median lines position at the boxes is getting closer to the 3rd quartiles (upper lines), while the 1st quartile line remains

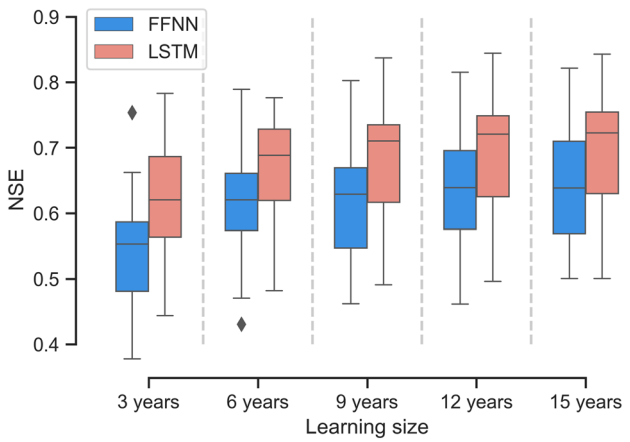

Fig. 6 Box plot of NSE for LSTM and FFNN prediction over the twenty catchments for the test period

unchanged. That indicates that for 25% of catchments, the grown sample of data in the learning procedure raised the performances of the model which can be due to the introduction of more informative data at these catchments. Moreover, since the NSE criteria are sensitive to extreme values, peaks of runoff may be present in the new data samples. For the FFNN, the sample distribution form of NSE values remains almost unchanged for different learning sizes. We can consider that the effect of the large data size is not the same for the FFNN prediction compared to the LSTM, at least in terms of extremes. Basing on the acceptability limits defined by Moriasi et al. (2007) that indicate a value of NSE higher than 0.65 as threshold, it can be considered that the deep model shows acceptable performances at 3 years of learning data size for around 30% of catchments. However, the FFNN performances explained by NSE values for all catchments are inferior to 0.60. These significant differences show the ability of the LSTM to be applied effectively despite the use of small data size in some situations compared to the FFNN. Moreover, for the LSTM, a leaning data length equal to 9 years is sufficient to reach acceptable performances for almost all catchments. For the FFNN, this is reached for less than 75% of basins at the same learning data size.

The graphical analysis of RMSE values (Fig. 7) shows that the LSTM model provides better performances compared to the FFNN at 14 tested watersheds (slight to important difference). There are some exceptions such as catchments 1, 3, 4, 5, 8 and 19, where the FFNN is outperforming the LSTM model at one or most of the training samples. Several assumptions may be the reasons of these cases, and one of the commonly reason observed in data-driven modeling is local minima trap. These exceptions can be explained by a poor training procedure of the LSTM for these situations leading to a bad generalization ability. In fact, the possibility to be trapped in local minima with bad generalization in such high complex relationship as rainfall runoff is possible even when using the advanced Adam algorithm with measures taken to avoid the overtraining problem (holdout method and dropout layer). Thus, there is a possibility to improve these performances by continuing the try-error procedure in these cases. Another reason of the failure origin of the LSTM model to outperform its benchmark may be the existence of an extreme event in training data. At some basins, we can notice that for both models, RMSE values increase while increasing training samples (such as seen in the basins 4 and 7 for, respectively, the LSTM and the FFNN models), which is contrary to what was expected. RMSE curve of the LSTM at catchment 4 decreases from 3 years of learning data (21.9 m^3/s) until 9 years (19.6 m^3/s), when this particular event is added to the training sample at 12 years, the RMSE curve is inversed (19.9 to 20.5 m^3/s at 15 years). In fact, by analyzing the runoff data (not

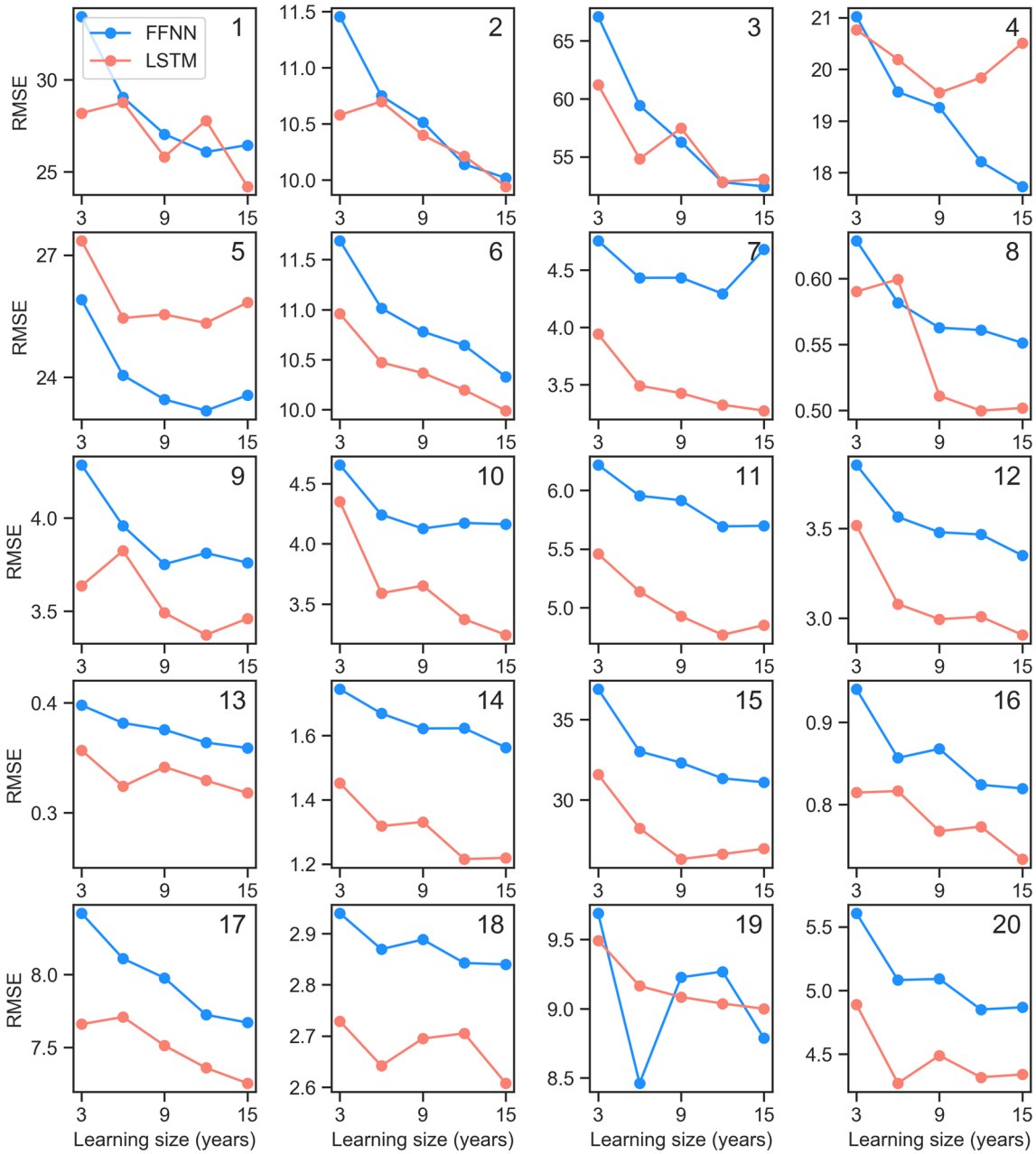

Fig. 7 Performance of LSTM and FFNN models for each catchment according to the RMSE at the period test (orange and blue colors for, respectively, LSTM and FFNN models)

shown in this study) of the catchment 4 served for the training process of both models, a presence of a high value of runoff (897.64 m^3/s) has been found when using 12 and 15 years of training data, while the rest of runoff data do not exceed the value of 414 m^3/s. The effect of these extreme events on the FFNN model behavior is less significant since in most cases, the RMSE curve of this latter is continuing to decrease (see basins 4 and 15). This finding indicates that the LSTM model is more sensitive to the existence of such event in training data than the classical FFNN. Compared to this latter which is not a recurrent network, the architecture of the LSTM consists on the use of the memory vector containing previous simulated runoff (varied between 30, 180 and 365 days in this study) to predict the runoff at each time step. Thus, the presence of exceptional event may significantly compromise the generalization of this model.

In most cases of the LSTM performances (RMSE), slopes are higher from 3 to 9 (or 3 to 12) years of learning data size and become flatter until 15 years (see basins 11 and 15).

These outcomes confirm NSE results where the improvement is less effective with learning data size higher than 9 years and almost insignificant after 12 years.

In order to show the effect of the learning data size on the hydrograph components of the LSTM, temporal variation of runoff (observed and simulated) at the catchment 8 for the last year of test period is presented in (Fig. 8). In general, more stability is reached by the increasing of learning data size. When dealing with the peak flow of 94 m^3/s, the LSTM showed a good ability to predict its timing at all the showed leaning data sizes (6 to 15 years). However, the model simulations have been comparatively underestimating this peak value. By using 15 years of learning data size, better estimate is observed, where the simulation reached 73 m^3/s. Regarding the medium values, they have been well simulated (magnitude, shape and timing) by the model at 9, 12 and 15 years of learning data sizes, except for the complex events that occurred in April and June. The LSTM

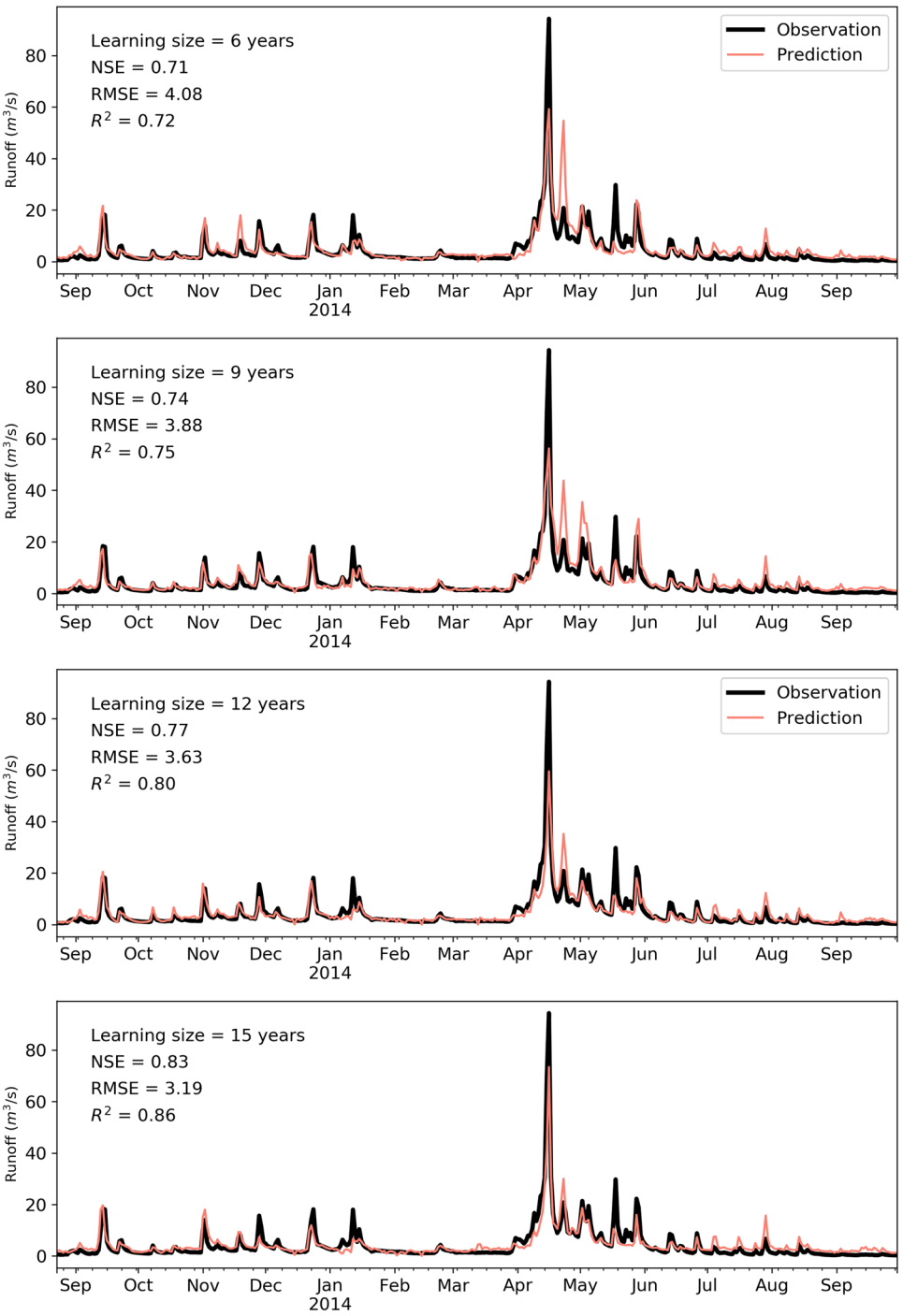

Fig. 8 Hydrograph of observed and LSTM predictions at the last year of test period with 6 to 15 years of training data size (catchment 8)

simulation at these events has been more reliable when using more than 9 years. By using 6 years of learning data size, the performance is acceptable with an NSE equal to 0.71 even if several medium flows are not well simulated. This matter can be justified by the small number of similar events in the time series, making the trained network unable to reproduce such events.

Among the evaluation of hydrological models, visual inspection still represents a fundamental step in model validation (Biondi et al. 2012). In this perspective, the scatter plot (Fig. 9) of both models (prediction against observed runoff data) at the twenty studied catchments has been carried out when using the full size of learning data. This plot is usually used and based on the fact that the agreement between predicted and observed values is considered as perfect when all the scatters are regrouped on the 1:1 line. Compared to the FFNN simulation, LSTM model showed better capability in predicting runoff with a R^2 ranged between 0.52 and 0.66 for seven catchments, while it exceeds 0.70 for the rest. These results confirm the previous findings in the literature (Hu et al. 2018; Kratzert et al. 2018) concerning the LSTM robustness in the rainfall runoff relationship.

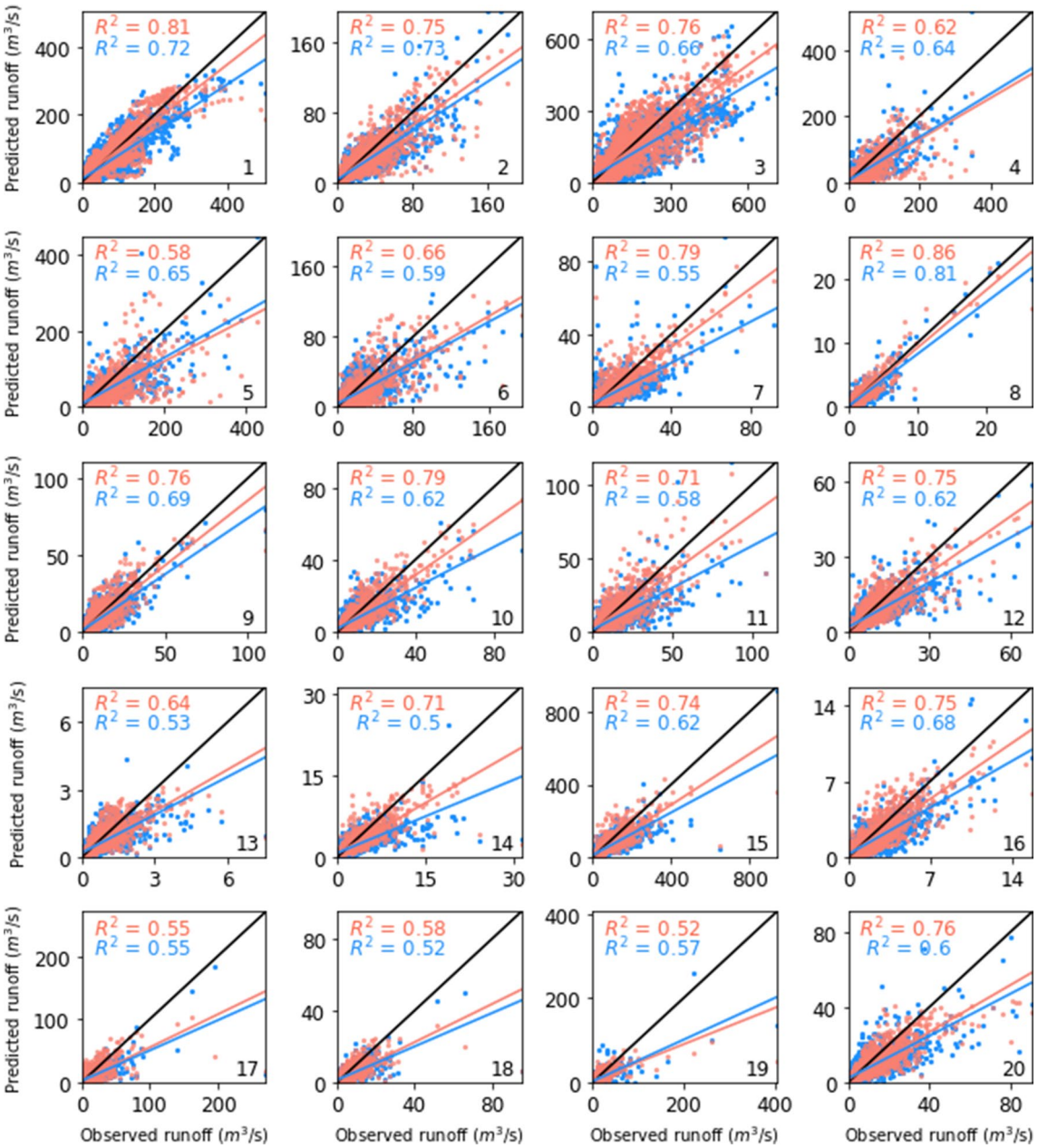

Fig. 9 Scatter plot of observed and predicted (FFNN and LSTM) runoff over the twenty catchments during the test period

The failure of the LSTM model in predicting runoff at some basins (17, 18 and 19) may be due to the highly skewed distribution of their runoff data (respectively, 7.6, 6 and 12). As is well known, high skewed data reflect the rarity of extreme events, which makes the generalization more difficult when encountering this kind of events.

In order to facilitate the interpretation (under- or overestimation) of these plots, linear trend lines of both models have been calculated (Fig. 9). It is interesting to note that the scatter diagrams clearly indicate that both networks tend to underestimate runoff at all the studied catchments. According to the scatter position, in most cases, it is largely due to the underestimation of peak flows rather than medium and low flows which are regrouped near the 1:1 line. This behavior of neural networks dealing with peak flows has been encountered in previous studies (Jimeno-Sáez et al. 2018; Panda et al. 2010).

Conclusion

In this work, a comparative study between deep learning (LSTM) and conventional network (FFNN) models was conducted. The purpose was to analyze the effect of the learning data size on the performance of the prediction accuracy. In this respect, five experiments based on varying the learning data size (3, 6, 9, 12 and 15 years) were carried out on twenty catchments with diverse hydrological conditions. Several efficiency criteria and graphical comparisons were used to judge the performance and evaluate the behavior of the studied models when using different training data size. Concerning the LSTM, it has been shown that a data length of 9 years is required for the training procedure to reach acceptable performances and 12 years for more efficient prediction. Since no significant improvements have been observed when using 15 years, we suggest that 12 years of training data are sufficient to capture most of the temporal hydrological variability. These thresholds concern similar meteorological and hydrological conditions of the studied catchments; thus, further applications of the present approach will enrich the study. On the other hand, the FFNN continues its improvement after 12 years of training data, but with low performances compared to the LSTM.

The deep learning model outperforms the benchmark model with less training data size. Using 3 years by the LSTM showed acceptable performances at 30% of basins, while the FFNN performance did not reach the threshold acceptability at almost all basins. Thus, when dealing with limited data, the deep learning model seems to be the best choice to implement due to its efficiency in capturing information with a minimum learning data size.

Despite being more reliable compared to the classical FFNN, the deep LSTM model shows some difficulties when an extreme event exists in the learning data. This was probably caused by the long sequence of cell memory which may affect negatively the generalization of the LSTM model in these situations. For future works, we suggest to select a training data with events having similar magnitudes and ignoring data period with extreme events. This solution may be difficult to reach in some situations due to the extreme event position. We suggest to perform a preprocessing operation dealing with outliers in order to avoid such problem.

Since the well-known uncertainties of satellite data products (Gebremichael and Hossain 2010; Hong et al. 2016), it is plausible that the use of the forcing satellite data (meteorological data) in the model may influence the results, especially the thresholds mentioned above. Accordingly, a database of ground-land measurements is needed to confirm these findings.

Acknowledgements This work was developed with the support of the Directorate General for Scientific Research and Technological Development DGRSTD in addition of the PRFU-MESRS Project (Code# A17N01UN230120180001).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Addor N, Nearing G, Prieto C, Newman AJ, Le Vine N, Clark MP (2018) Selection of hydrological signatures for large-sample hydrology. Earth arXiv: 12 Feb 2018 Web

Aggarwal CC (2018) Neural networks and deep learning. Springer 10:978–983

Anctil F, Perrin C, Andréassian V (2004) Impact of the length of observed records on the performance of ANN and of conceptual parsimonious rainfall–runoff forecasting models. Environ Model Softw 19:357–368

Ayzel G (2019) Does deep learning advance hourly runoff predictions? In: Sergey I Smagin, Alexander A Zatsarinnyy (eds): 5th International conference information technologies and high-performance computing (ITHPC-2019), Khabarovsk, Russia: CEUR Workshop Proceedings

Biondi D, Freni G, Iacobellis V, Mascaro G, Montanari A (2012) Validation of hydrological models: conceptual basis, methodological approaches and a proposal for a code of practice. Phys Chem Earth Parts A/B/C 42:70–76

Brath A, Montanari A, Toth E (2004) Analysis of the effects of different scenarios of historical data availability on the calibration of a spatially-distributed hydrological model. J Hydrol 291:232–253

Chen SM, Wang YM, Tsou I (2013) Using artificial neural network approach for modelling rainfall–runoff due to typhoon. J Earth Syst Sci 122:399–405

Chen J, Zeng G-Q, Zhou W, Du W, Lu KD (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Convers Manag 165:681–695

Gebremichael M, Hossain F (2010) Satellite rainfall applications for surface hydrology. Springer, Berlin

Ghimire S, Deo RC, Raj N, Mi J (2019) Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. Appl Energy 253:113541

Graves A, Mohamed A-R, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 6645–6649

Guermoui M, Rabehi A, Benkaciali S, Djafer D (2016) Daily global solar radiation modelling using multi-layer perceptron neural networks in semi-arid region. Leonardo Electron J Pract Technol 28:35–46

Gupta VK, Sorooshian S (1985a) The automatic calibration of conceptual catchment models using derivative-based optimization algorithms. Water Resour Res 21:473–485

Gupta VK, Sorooshian S (1985b) The relationship between data and the precision of parameter estimates of hydrologic models. J Hydrol 81:57–77

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780

Hong Y, Zhang Y, Khan SI (2016) Hydrologic remote sensing: capacity building for sustainability and resilience. CRC Press, Boca Raton

Hu C, Wu Q, Li H, Jian S, Li N, Lou Z (2018) Deep learning with a long short-term memory networks approach for rainfall–runoff simulation. Water 10:1543

Jeong DI, Kim YO (2005) Rainfall–runoff models using artificial neural networks for ensemble streamflow prediction. Hydrol Process Int J 19:3819–3835

Jimeno-Sáez P, Senent-Aparicio J, Pérez-Sánchez J, Pulido-Velazquez D (2018) A comparison of SWAT and ANN models for daily runoff simulation in different climatic zones of Peninsular Spain. Water 10:192

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980

Kratzert F, Klotz D, Brenner C, Schulz K, Herrnegger M (2018) Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrol Earth Syst Sci 22:6005–6022

Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. Q Appl Math 2:164–168

Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. J Soci Ind Appl Math 11:431–441

Merz R, Parajka J, Blöschl G (2009) Scale effects in conceptual hydrological modeling. Water Resour Res 45:W09405

Moriasi DN, Arnold JG, Van Liew MW, Bingner RL, Harmel RD, Veith TL (2007) Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Trans ASABE 50:885–900

Newman AJ et al (2015) 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. Hydrol Earth Syst Sci 19:209

Panda RK, Pramanik N, Bala B (2010) Simulation of river stage using artificial neural network and MIKE 11 hydrodynamic model. Comput Geosci 36:735–745

Perrin C, Oudin L, Andreassian V, Rojas-Serna C, Michel C, Mathevet T (2007) Impact of limited streamflow data on the efficiency and the parameters of rainfall–runoff models. Hydrol Sci J 52:131–151

Remesan R, Mathew J (2015) Hydroinformatics and data-based modelling issues in hydrology. In: Hydrological data driven modelling. Springer, pp 19–39

Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747

Solaimani K (2009) Rainfall-runoff prediction based on artificial neural network (a case study: Jarahi watershed). Am–Eurasian J Agric Environ Sci 5:856–865

Srivastava S, Lessmann S (2018) A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. Sol Energy 162:232–247

Thornton PE, Thornton MM, Mayer BW, Wilhelmi N, Wei Y, Devarakonda R, Cook R (2012) Daymet: daily surface weather on a 1 km grid for North America, 1980–2008. Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center for Biogeochemical Dynamics (DAAC)

Tokar AS, Johnson PA (1999) Rainfall–runoff modeling using artificial neural networks. J Hydrol Eng 4:232–239

Wang H, Yang Z, Yu Q, Hong T, Lin X (2018) Online reliability time series prediction via convolutional neural network and long short term memory for service-oriented systems. Knowl Based Syst 159:132–147

Wöllmer M, Kaiser M, Eyben F, Schuller B, Rigoll G (2013) LSTM-modeling of continuous emotions in an audiovisual affect recognition framework. Image Vis Comput 31:153–163

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.