SSRG International Journal of Geo-informatics and Geological Science
ISSN: 2393–9206 / https://doi.org/10.14445/23939206/IJGGS-V10I1P103

Volume 10 Issue 1, 29-35, Jan-Apr 2023

Original Article

Modeling River Discharge using Deep Learning in the Ouémé catchment at Savè outlet (Benin, West Africa)

Zohou Pierre Jérôme[1,2,3,4], Biao Iboukoun Eliézer[4,5], Aoga John[1,3], Houessou Oscar[5], Alamou Adéchina Eric[4,5], Eugène C.Ezin[1,3]

[1]*Université d'Abomey-Calavi (UAC), Bénin.*
[2]*Chaire Internationale en Physique Mathématique et Applications (CIPMA-Chaire UNESCO), UAC, Bénin.*
[3]*Laboratoire de Recherche en Sciences Informatiques et Applications (LRSIA), UAC, Bénin*
[4]*Laboratoire d'Hydrologie Appliquée (LHA), UAC, Bénin*
[5]*Université Nationale des Sciences, Technologies, Ingénierie et Mathématiques (UNSTIM), Abomey, Bénin*

Received: 22 February 2023 Revised: 25 March 2023 Accepted: 08 April 2023 Published: 20 April 2023

Abstract *- This paper presents a modeling approach based on Artificial Neural Networks (ANNs) in the Ouémé river catchment at Savè. To do this, we used precipitation data as input over the period 1965 -2010 to simulate river discharge in the study area by using two ANNs models such as the Long Short Term Memory (LSTM) and Recurrent Gate Networks (GRU) models. Indeed, the description of the stochastic nature of the data is better presented today by ANNs models than the statistical models. We compared the performance of these two models based on different evaluation criteria. The predictions made using these models show a strong similarity between the observed and simulated flows. The deep learning models gave good results. Indeed, in calibration and validation, the Nash Sutcliffe Efficiency (NSE) and the coefficient of determination (R^2) are very close to one (calibration: R^2= 0.995, NSE= 0.991, and RMSE= 0.18; validation: R^2 = 0.975, NSE= 0.971, and RMSE= 0.41). This good performance of LSTM and GRU confirms the importance of models based on Artificial Intelligence in modeling hydrological phenomena for better decision-making.*

Keywords *- Artificial Neural Networks, Modeling, Ouémé catchment at Savè, Long Short Term Memory, Gated Recurrent Unit.*

1. Introduction

Precipitation is a natural phenomenon and is generally the largest contributor to the water balance in a watershed. They comprise drizzle, ice, frost, snow, hail, sleet, and rain. However, in West Africa, specifically in Benin, rain feeds the water tables and the various rivers [8]. However, excessive rainfall leads to natural disasters such as flooding. It is, therefore, necessary to better control the phenomenon of precipitation, and this requires its modelling. Over the past few decades, fully data-driven (empirical) models have begun to emerge with breakthroughs in new deep-learning methods in runoff prediction [11]. These breakthroughs were mainly made possible by the availability of large volumes of water-related data. We propose using recurrent neural networks models such as LSTM and GRU to model the rainfall-runoff relationship. To achieve this, we will optimize the hyperparameters of the models, simulate the river discharge at the outlet of the catchment area and finally evaluate the performance of the recurrent neural network models.

2. Materials and Methods

The Ouémé is a river that covers at Bonou, the most advanced station before the Delta, an area of 46,990km^2. It rises at the foot of Atacora, in the Djougou region, crosses Benin towards the coast, and flows into Lake Nokoué, just north of Cotonou (Fig. 1). It is thus the longest river in Benin, draining more than a third of the territory alone. The Ouémé basin at Savè (09°12'N; 02°16'E) is the area whose data are used in this project. Its natural outlet, located a few kilometers downstream from the confluence of the Ouémé with the Yérou-Maro, is the Bétérou station, created in 1952; the area covered by the Ouémé in Bétérou is 10,475km^2.

Precipitation data used comes from Météo-Bénin (National Meteorological Agency of Benin), while the National Directorate of Water (DG-Eau) provides the river discharge data. The study area contains seven rainfall stations (Savè, Ouesse, Kokoro, Tchaourou, Bassila, Penessoulou, Toui) covering the period from 1965 to 2010.

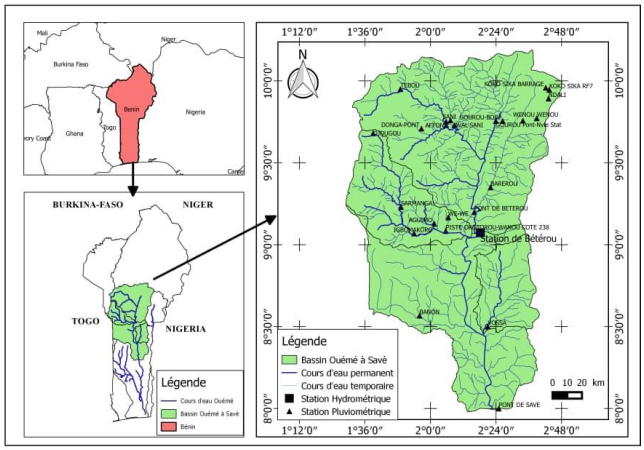

Fig. 1 Geographical location of the Ouémé basins in Savè

Our methodology for modeling the rainfall-runoff relationship using LSTM and GRU-type recurrent neural networks is based on four steps.

2.1. Data Preprocessing

Before loading the data into the LSTM and GRU models, a few transformations were applied, such as data normalization and transforming time series into supervised learning series. We use normalization and standardization methods to reduce the complexity of LSTM and GRU models [16].

Normalization scales each input variable (precipitation and evapotranspiration) separately in the range 0 -1, the range of floating point values where we have the most precision.

$$X_{normalize} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Standardization, like normalization, scales the output variable (rate) by subtracting the mean (called centering) and dividing by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of one [22].

2.1.1. Split the Dataset

Our hydrometeorological data is divided into three main parts to ensure the training, validation, and testing of recurrent neural network models (LSTM & GRU).

Table 1. Dataset split

Phase	Percentage	Period
Training set	60%	1965-2010
Validation set	20%	
Test set	20%	

2.1.2. Training Set

A first data set will be used to train the models. This set covers 60% (01-01-1965 to 12-31-1992) of the dataset. This data set allows learning the different weights of the neurons constituting our network.

2.1.3. Validation Set

A second data set will be used to validate the model parameters (validation set). This set represents 20% of our database (01-01-1993 to 12-31-2001). This data sample provides an unbiased evaluation of the model fit on the training data set while adjusting the model hyperparameters.

2.1.4. Test Set

A third data set will be used to test the real performance of our model. This dataset also represents 20% of our database (01-01-2002 to 12-31-2010). This is the test sample. It was used only after the model was fully trained (using the training and validation sets). It is used to provide an unbiased assessment of the fit of the final model on the training dataset.

2.2. Construction and Validation of Forecasting Models

This phase consists of training the LSTM and GRU models and validating their performance by optimizing some of their hyper-parameters.

An artificial neural network is like an assembly of identical structural elements called cells (or neurons) interconnected like the nervous system cells of vertebrates. The information in the network propagates from one layer to another, and they are said to be of a "feed-forward" type [23]. We distinguish three types of layers:

2.2.1. Input Layers

The neurons in this layer receive the input values from the network and pass them on to the hidden neurons.

Each neuron receives a value, so it does not sum;

2.2.2. Hidden Layers

Each neuron of this layer receives information from several previous layers, performs the summation weighted by the weights, and then transforms it according to its activation function, which is generally a sigmoid function [21]; it is the most suitable for the hydrological model. It then sends this response to neurons of the next layer;

2.2.3. Output Layers

These play the same role as the hidden layers, the only difference between these two types of layers is that the output of the neurons of the output layer is not linked to any other neuron

2.3. Artificial Neural Networks

The input variable X of our artificial neural network is composed of precipitation and evapotranspiration data (x_1 : Precipitation; x_2: ETP).

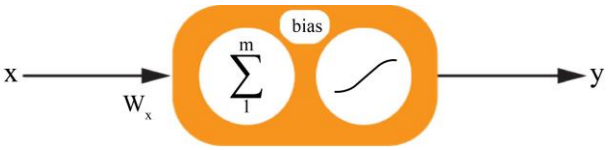

Fig. 2 Cell of Artificial Neural Network

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots \\ x_1^{(m)} & x_1^{(m)} \end{bmatrix} \in \mathbb{R}^{m\times 2} \quad (2)$$

With m, the number of observations.

In an artificial neuron, the input variable goes through an aggregation step and then an activation step that allows activating or not the neuron. The first aggregation step consists in making the weighted sum of the inputs and the weights of the neurons (W) to which we add a bias (b).

$$Z = XW + b$$

$$\mathrm{Z} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots \\ x_1^{(m)} & x_1^{(m)} \end{bmatrix} . \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} b \\ b \\ \vdots \\ b \end{bmatrix} \quad (3)$$

$$\mathrm{Z} = \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(m)} \end{bmatrix} = \begin{bmatrix} w_1 x_1^{(1)} + w_2 x_2^{(1)} + b \\ w_1 x_1^{(2)} + w_2 x_2^{(2)} + b \\ \vdots \\ w_1 x_1^{(m)} + w_2 x_2^{(m)} + b \end{bmatrix} \quad (4)$$

The second step consists in passing the value of Z in the activation function (sigmoid function, [11]), which allows activating or not the neuron.

$$a^{(i)} = \sigma\left(z^{(i)}\right) = \frac{1}{1+e^{-z^{(i)}}}$$

$$\mathrm{A} = \begin{bmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(m)} \end{bmatrix} = \sigma\left(\begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(m)} \end{bmatrix}\right) \quad (5)$$

In summary, in an Artificial Neural Network, we have:

$$\mathrm{y} = \sigma(\mathrm{XW} + \mathrm{b})$$

2.4. Recurrent Neural Networks

A recurrent neural network is an artificial neural network with recurrent connections. A recurrent neural network consists of interconnected units (neurons) interacting non-linearly, for which there is at least one cycle in the structure. The units are connected by arcs (synapses) which have a weight. The output of a neuron is a nonlinear combination of its inputs. Recurrent neural networks are suitable for time series analysis.

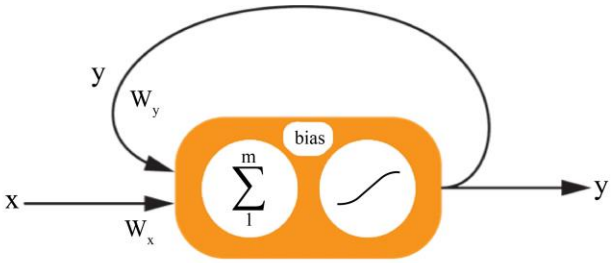

Fig. 3 Cell of a Recurrent Neural Network

$$y_t = \sigma\left(\mathrm{X}W_X + y_{t-1} * W_y + \mathrm{b}\right)$$

A Long Short Term Memory (LSTM) neural network [6] is the most widely used recurrent neural network architecture in practice that addresses the gradient vanishing problem. The idea associated with LSTM is that each computational unit is linked to a hidden state h and a state c of the cell, which acts as a memory. The transition from $c_{(t-1)}$ to c_t is done by a constant gain transfer equal to one [24]. In this way, errors are propagated to previous steps (up to 1000 steps in the past) without any gradient disappearance phenomenon. The state of the cell can be modified through a gate that allows or blocks the update (input gate). Similarly, a gate controls whether the cell state is communicated at the output of the LSTM unit (output gate). The most common version of LSTMs also uses a forget gate to reset the cell state.

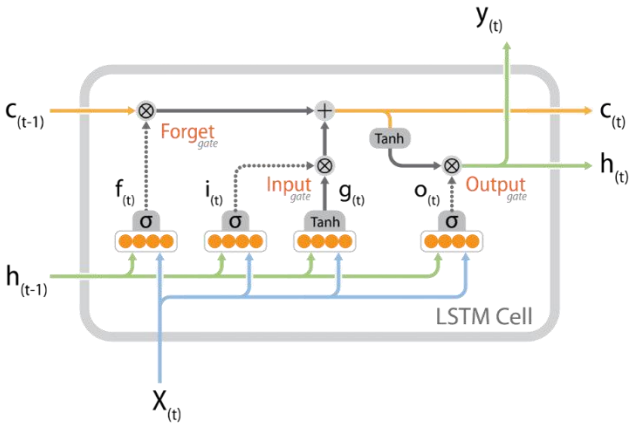

Fig. 4 LSTM network architecture

The different formulas for each gate (forget gate, input gate, output gate) are presented below.

$$f_{(t)} = \sigma\left(W_{xt}^T . X_{(t)} + W_{hf}^T . h_{(t-1)} + b_f\right)$$

$$i_{(t)} = \sigma\left(W_{xi}^T . X_{(t)} + W_{hi}^T . h_{(t-1)} + b_i\right)$$

$$g_{(t)} = tanh\left(W_{xg}^T . X_{(t)} + W_{hg}^T . h_{(t-1)} + b_g\right)$$

$$o_{(t)} = \sigma\left(W_{xo}^{T}.X_{(t)} + W_{ho}^{T}.h_{(t-1)} + b_o\right)$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)}$$

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh\left(c_{(t)}\right)$$

2.5. Gated Recurrent Unit (GRU) Network

A Gated Recurrent Unit (GRU) network [5] is a variant of LSTMs introduced in 2014. GRU networks have performance comparable to LSTMs for time series prediction. A GRU unit requires fewer parameters to learn than an LSTM unit. A neuron is now associated with only one hidden state, and the gates of entering and forgetting the hidden state are merged [27]. The output gate is replaced by a reset gate.

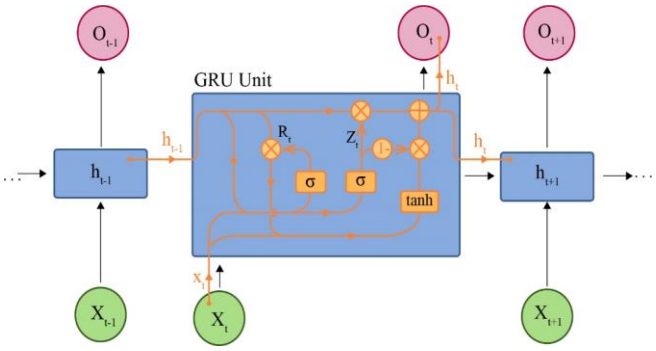

Fig. 5 GRU Network Architecture

2.6. Optimization of Hyperparameters of LSTM and GRU models

While constructing recurrent neural network models, we are faced with the choice of hyperparameters. Indeed, a hyperparameter is a parameter whose value is used to control the learning process. They are adjustment parameters of the machine learning algorithms. It is known that the hyperparameters of an artificial neural network have an influence on the performance of the model, so the number of units in the LSTM layers, the batch size, and the learning rate of the optimizer are selected as optimization objects. Optimizing the hyperparameters of an LSTM or GRU model involves performing a search to discover the set of model configuration arguments that result in the best model performance on a specific data set. The hyperparameters to be optimized during the training phase of LSTM and GRU models are:

2.6.1. Number of Hidden Units by Layer

These must also be chosen reasonably to find a trade-off between high bias and high variance. Again, this depends on the size of the data used for training.

2.6.2. Learning Rate

This is a hyperparameter that plays on the speed of the gradient descent: a more or less important number of iterations is necessary before the algorithm converges, i.e. before optimal learning of the network is achieved.

2.6.3. Batch Size

Several samples that will be transmitted to the network at one time. It is also commonly referred to as a mini lot. If the batch size is smaller, the patterns would be less repetitive and hence convergence would become difficult. If the batch size is large, the learning is slow because it is only after many iterations that the batch size will change.

2.6.4. Number of Epochs

The number of epochs is the number of times all the training data are presented to the model.

It plays an important role in how well the model fits the training data.

The architectures of the recurrent neural network models developed consist of three layers, namely:

- An input layer made up of vectors comprising the values of the input variables (precipitation and evapotranspiration);
- A hidden layer (LSTM or GRU) composed of 100 units;
- An output layer composed of a neuron that predicts the value of the flow.

The optimizer used is the Adam optimizer. [26] list the attractive benefits of using Adam on non-convex optimization problems, as follows: Straightforward to implement; computationally efficient; little memory requirements; invariant to diagonal rescale of the gradients; well suited for problems that are large in terms of data and/or parameters.

The hyper-parameters have intuitive interpretation and typically require little tuning. The loss function chosen is the root mean square error. For the training phase of the LSTM and GRU models, the number of epochs was set to 100 to have the same scale of comparison between the models. Model evaluation was performed using the test dataset. We evaluated the models by analyzing the curve of the loss function on the number of epochs [15]

2.7. Model Performance Evaluation

To assess the performance of the models, the Nash Stutcliffe efficiency (NSE), the coefficient of determination (R^2), and the root mean squared error (RMSE) are statistical methods often used to compare predicted values to observed values.

2.8. Performance Evaluation Criteria

2.8.1. Nash-Sutcliffe criterion (or Nash criterion)

This metric (3) expresses the proportion of the variance of flows explained by the hydrological model. According to

[14], in hydrological applications and depending on the type of river, the Nash criteria are acceptable between 0.60 and 0.70 and excellent if they are greater than 0.90.

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(Q_{i,obs} - Q_{i,sim}\right)^2}{\sum_{i=1}^{n}\left(Q_{i,obs} - \bar{Q}\right)^2} \quad (6)$$

Where $Q_{i,obs}$, $\overline{Q}$, and n stand respectively for the observed river discharge at day i, the simulated discharge at day i, the average river discharge, and n, the total number of days. The closer the value of the criterion is to 1, the more accurate the model is. A negative criterion value occurs when the observed mean is a better prediction than the model, in other words, when the residual variance is larger than the data variance.

2.8.2. Coefficient of determination R^2

The value of R^2 describes the proportion of the variance of the observed discharges compared to the simulation discharges.

$$R^2 = \frac{\sum_{i=1}^{n}\left(Q_{i,sim} - \bar{Q}_{i,obs}\right)^2}{\sum_{i=1}^{n}\left(Q_{i,obs} - \overline{Q}\right)^2} \quad (7)$$

Authors such as (Moriasi, 2015) suggest that any R^2 value greater than 0.5 for comparisons of daily flows is an acceptable threshold in hydrological simulation.

2.8.3. Root Mean Square Error (RMSE)

The square root of the means square error RMSE is one of the most widely used evaluation criteria in forecasting research [25], which gives a quantitative indication of the overall error produced by the model. This criterion determines the deviation of the simulated value from the observed mean.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Q_{i,obs} - Q_{i,sim}\right)^2} \quad (8)$$

The closer the mean squared error value is to 0, the better the simulation result is.

3. Results and Discussion

3.1. Models Training and Validation

We can see in Fig. 6 the evolution of the loss function (Loss) during the training of the two models of recurrent neural networks (LSTM and GRU). It can be seen that the error during the training and test phases converges towards zero just after a few tens of epochs. One can deduct from this fact that models based on machine learning require very few computing resources while allowing them to have very good results.

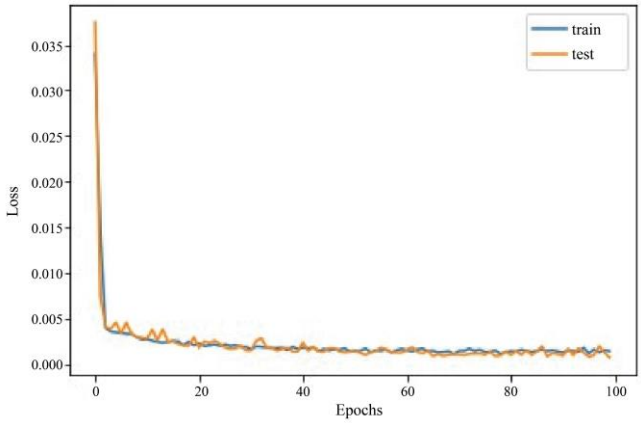

Fig. 6 Error evolution curve during the test training phases

3.2. Hyperparameters Tuning Values

Fig. 7 and Fig. 8 below show the values of the selected hyperparameters after hyperparameter optimization.

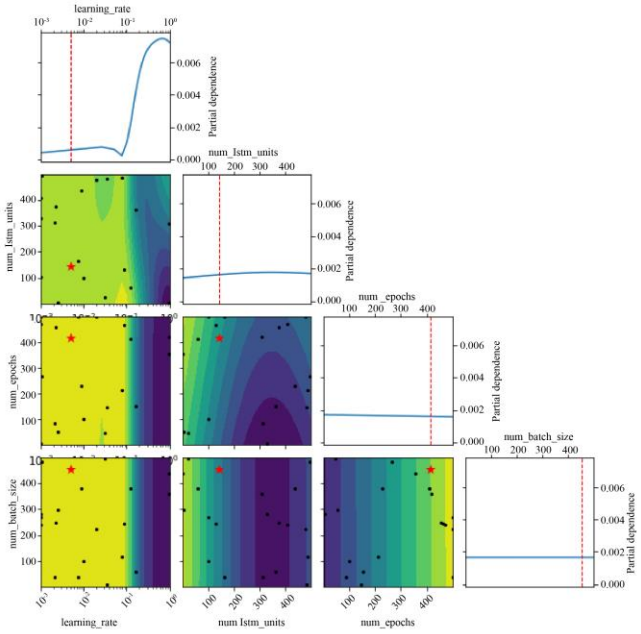

Fig. 7 Value of LSTM model hyperparameters

Table 2. Hyperparameter value

Models	Learning rate	Number of units	Number of epochs	Batch size
LSTM	0.0051	144	415	454
GRU	0.01	222	294	144

Both recurrent neural network models perform better with lower learning rates and several units smaller than 300. The number of epochs and the batch size has less influence on the models, although a higher number of epochs slightly improves simulations. The models obtained good results in calibration and validation. After the training phase of the LSTM and GRU models, we obtain almost very good models. Indeed, the performances obtained in the NSE and R^2 tests are very close to one. Similarly, the mean square error is close to zero (Table 3).

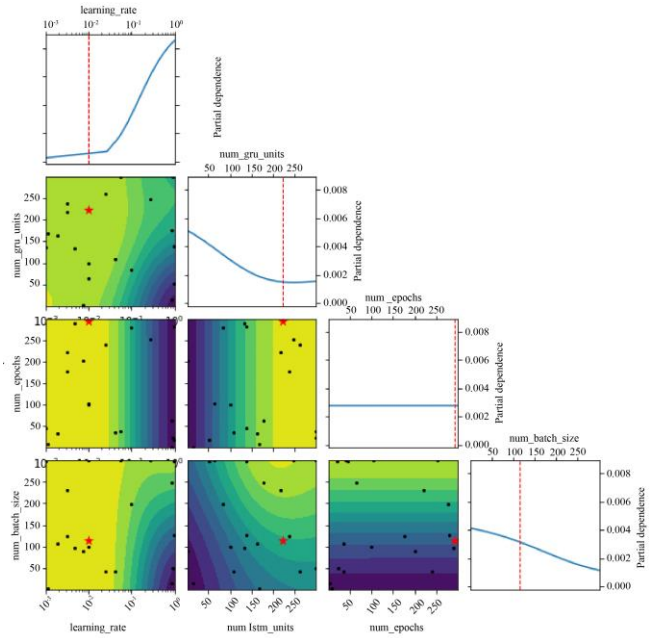

Fig. 8 Value of GRU model hyperparameters

These various values allowing obtaining the best performances during the simulation of the flows (Table 2).

Table 3. Results of models in calibrations of the model calibrations

Metrics	LSTM	GRU
Coefficient of determination R^2	0.994	0.995
Nash-Sutcliffe Efficiency (NSE)	0. 989	0.991
Root Mean Square Error (RMSE)	0.20	0.18

3.3. Simulation with LSTM and GRU models

After training, the validation of the two models was carried out with the test data ranging from the period 2002 to 2010.

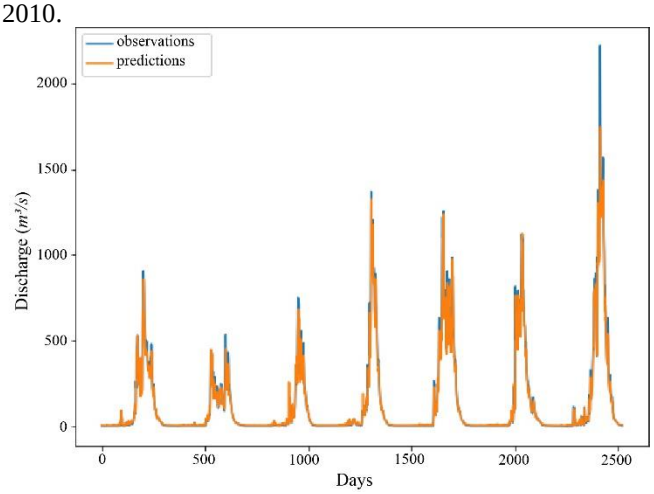

Fig. 9 Discharge simulated with LSTM neural network

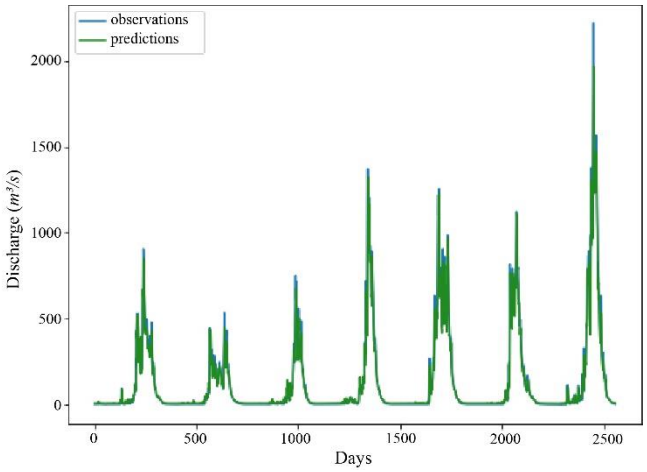

Fig. 10 Discharge simulated with LSTM neural network

Both models performed very well on this test data. It comes from these figures that the models (LSTM and GRU) present very good results in calibration. Indeed, the LSTM and GRU obtained the same scores of 0.971 for the NSE test and 0.975 as the value for R^2 in validation (Table 4). The LSTM and GRU simulation results are the same based on the obtained R^2 and NSE performance criteria. However, with the RMSE criterion, it can be seen that the LSTM and GRU models give respectively 0.44 and 0.41. We can therefore deduce that the GRU model provides slightly better results than the LSTM model because the mean square error of the latter is closest to zero. These calibration performances produced by LSTM and GRU models well exceed the acceptable threshold in rainfall-runoff modeling at the daily scale proposed by [12], which is 0.5.

Table 4. Evaluation of the performances of the two models

Metrics	LSTM	GRU
Coefficient of determination R^2	0.975	0.975
Nash-Sutcliffe Efficiency (NSE)	0. 971	0.971
Root Mean Square Error (RMSE)	0.44	0.41

Table 5. Performance obtained by hydrological models

Metrics	Calibration	Validation
R^2	0.86	0.83
NSE	0.82	0.81

Table 5 presents the results obtained by [2] on the Ouémé basin in Savè using a hydrological model. It can be seen that the performance achieved with recurrent neural networks largely exceeds that obtained by other authors using classical hydrological models.

4. Conclusion

This work aimed to assess the potential for using recurrent neural networks in rainfall-runoff modeling in the Ouémé basin at Savè over the period 1965 to 2010. We used two recurrent neural network models (LSTM and GRU), leading us to very conclusive results. The simulations obtained show a strong similarity between the observed and simulated flows. This demonstrates the effectiveness of artificial intelligence-based models in hydrological modeling. A combination of the LSTM and GRU models to set up an overall model may better improve the simulation.

References

[1] Agon, Orel, "Rainfall Variability and Impacts on Surface Water Resources in the Zou Basin at Atcherigbe," *International Journal of Progressive Sciences and Technologies*, 2020.

[2] A. Alamou, "*Application of the Least Action Principle to Rainfall Flow Modeling*," Unique Doctoral Thesis, University of Abomey-Calavi, p. 231, 2016.

[3] Balduíno César Mateus et al., "Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press," *Energies*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Eliézer Iboukoun Biao et al., "*Improving Rainfall-Runoff Modelling through Uncertainties' Control under Increasing Climate Variability in the Oueme*," Doctoral Thesis, University of Abomey-Calavi, 2015.

[5] Junyoung Chung et al., "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv*, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[6] Sepp Hochreiter, and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, pp. 1735-1780, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[7] M.A. Manjunatha Swamy et al., "Evaluating the Impacts of Climate Change on Thunga River Basin Karnataka," *SSRG International Journal of Agriculture & Environmental Science*, vol. 8, no. 5, pp. 1-8, 2021. [CrossRef] [Publisher Link]

[8] Y. B. Koffi, Study of the Calibration, Validation and Performance of Formal Neural Networks Based on Hydro-Climatic Data from the Bandama Blanc Watershed in Ivory Coast, 2014.

[9] N. Laborde, *Towards a New Approach for Determining the Siltation Rate of Dams*, 1998.

[10] S. Lallahem, "Structure and Hydrodynamic Modeling of Two Groundwaters: Application to the Chalky Aquifer of the Northern Border of the Paris Basin," Lille: Geological Society of the North, 2023.

[11] M.-A. Boucher, J. Q "Data Assimilation for Streamflow Forecasting," *Water Ressources Research*, 2020.

[12] Daniel N. Moriasi, "Hydrologic and Water Quality Models: Key Calibration and Validation," *American Society of Agricultural and Biological* Engineers, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] K. P. Sudheer, A. K. Gosain, and K. S. Ramasastri, "A Data-Driven Algorithm for Constructing Artificial Neural Network Rainfall-Runoff Models," *Hydrological Processes, Hydrological Processes*, vol. 16, no. 6, pp. 1325-1330, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mohamed Samir Toukourou, *Application of Machine Learning to Flash Flood Forecasting*, ParisTech Online Theses, 2015.

[15] A.Vannieuwenhuyze, *Machine Learning and Deep Learning by doing*, 2019.

[16] Xuan-Hien Le et al., "Comparison of Deep Learning Methods for Hydrological Forecasting in a Large-Scale Basin," *Remote Sensing of Environment*, vol. 251, p. 111631, 2020.

[17] S. Sadeghian et al., "Modeling and Forecasting River Flow Using Hybridized Deep Learning Methods," *Journal of Hydrology*, vol. 569, pp. 594-606, 2019.

[18] Télesphore Cossi Nounangnonhou et al., "*Modelling and Prediction of Ouémé (Benin) River Flows by 2040 Based on Gr2m Approach*," Accra Technical University Library, 2018. [Google Scholar] [Publisher Link]

[19] V. Banupriya, and S. Anusuya, "Improving Classification of Retinal Fundus Image Using Flow Dynamics Optimized Deep Learning Methods," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 12, pp. 39-48, 2022. [CrossRef] [Publisher Link]

[20] Poornima, S. Pushpalatha, M. "Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units," *Atmosphere*, vol. 10, p. 668, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Zhongrun Xiang, Jun Yan, and Ibrahim Demir, "A Rainfall-Runoff Model with LSTM-Based Sequence-To-Sequence Learning," *Water Resources Research*, vol. 56, p. e2019WR025326, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Dechao Sun et al., "Prediction of Short-Time Rainfall Based on Deep Learning," *Mathematical Problems in Engineering*, vol. 2021, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] S. Riad, J. Mania, L. Bouchaou, Y. Najjar, "Rainfall-Runoff Model Usingan Artificial Neural Network Approach," *Mathematical and Computer Modelling*, vol. 40, pp. 839-846, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[24] John Abbot, Jennifer Marohasy, "Input Selection and Optimisation for Monthly Rainfall Forecasting in Queensland, Australia, Using Artificial Neural Networks," *Atmospheric Research*, pp. 166-178, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[25] Christian W. Dawson, and Robert Wilby, "An Artificial Neural Network Approach to Rainfall-Runoff Modelling," *Hydrological Sciences Journal*, vol. 43, no. 1, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[26] Diederik Kingma, and Jimmy Ba, "A Method for Stochastic Optimzation," *arXiv*, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] Longzhang Fang, and Dongguo Shao, "Application of Long Short-Term Memory (LSTM) on the Prediction of Rainfall-Runoff in Karst Area," Frontiers in Physics, vol. 9, p. 790687, 2022. [CrossRef] [Google Scholar] [Publisher Link]