

DOI: 10.24850/j-tyca-2024-06-03

Articles

Application of artificial neural networks to the modeling of rain-runoff in the Chancay Lambayeque river basin

Aplicación de redes neuronales artificiales a la modelación de lluvia-escorrentía en la cuenca del río Chancay Lambayeque

Lourdes Ordoñez¹, ORCID: <https://orcid.org/0000-0002-6028-9497>

Sócrates Muñoz², ORCID: <https://orcid.org/0000-0003-3182-8735>

Percy Tineo³, ORCID: <https://orcid.org/0000-0002-6877-6302>

Iván Mejía⁴, ORCID: <https://orcid.org/0000-0002-0007-0928>

¹Universidad Señor de Sipán; Faculty of Engineering, Architecture and Urban Planning; Professional School of Civil Engineering. Chiclayo, Lambayeque, Peru, orimarachinlour@uss.edu.pe

²Universidad Nacional Toribio Rodríguez de Mendoza de Amazonas, Chachapoyas, Peru, socrates.munoz@untrm.edu.pe

³Universidad Nacional de Ingeniería, Faculty of Civil Engineering, Lima, Lima, Peru, percy.tineo.p@uni.pe

⁴Universidad Señor de Sipán; Faculty of Engineering, Architecture and Urban Planning; Professional School of Systems Engineering. Chiclayo, Lambayeque, Perú, hmejiac@uss.edu.pe

Corresponding author: Sócrates Pedro Muñoz-Pérez,
socrates.munoz@untrm.edu.pe

Abstract

Between the months of December to April, regions of northern Peru, including Lambayeque, are affected by maximum extreme events, wreaking havoc on homes, flooding crop fields, collapsing hydraulic works, and the most irreparable loss of human lives. In this line, the objective of this research was to apply Artificial Neural Networks to rain-runoff modeling in a basin in northern Peru, namely, the Chancay Lambayeque river basin belonging to the Pacific slope. For this purpose, records of precipitation and flows of 30 years (hydrological normal) were collected from 12 hydrometeorological stations belonging to the basin and neighboring it. Thus, applying a model of Long and Short Term Memory Networks (LSTM) we proceeded to model the rain, seeking to follow the behavior of the flows observed in the Racarrumi hydrometric station, with 80 % of the information the model was trained and with 20 % it was validated. In short, it was obtained that in the modeling validation stage, the Nash coefficient was 0.93, corresponding to the qualifier "very good".

Keywords: Basin, precipitation, flow, artificial neural networks, hydrometeorological stations.

Resumen

Entre los meses de diciembre a abril, regiones del norte del Perú, entre ellas Lambayeque, se ven afectados por eventos extremos máximos, ocasionando estragos en viviendas, inundación de campos de cultivo, colapso de obras hidráulicas y, lo más irreparable, pérdida de vidas humanas. En esa línea, el objetivo de la presente investigación fue aplicar redes neuronales artificiales al modelamiento de lluvia-escorrentía en una cuenca del norte de Perú, en específico, en la cuenca del río Chancay Lambayeque, perteneciente a la vertiente del Pacífico. Para ello se recopilaron registros de precipitación y caudales de 30 años (normal hidrológica), de 12 estaciones hidrometeorológicas pertenecientes tanto a la cuenca como aledañas a ésta. Así, aplicando un modelo de redes de memoria a largo y corto plazo (LSTM), se procedió a modelar la lluvia, buscando seguir el comportamiento de los caudales observados en la estación hidrométrica Racarrumi; con un 80 % de la información se entrenó al modelo y con un 20 % se validó. En suma, se obtuvo que en la etapa de validación del modelamiento, el coeficiente de Nash fue de 0.93, correspondiéndole el calificativo de “muy bueno”.

Palabras clave: cuenca, precipitación, caudales, redes neuronales artificiales, estaciones hidrometeorológicas.

Received: 15/12/2022

Accepted: 07/07/2023

Published Online: 14/07/2023

Introduction

Flow behavior during rainfall events occurs as a nonlinear process, and runoff modeling is one of the key challenges in the field of hydrology, especially in watersheds where topographic data are not available (Fan *et al.*, 2020). Research on rainfall-runoff modeling contributes to risk reduction, counteracting damages, minimizing loss of human lives and reducing the impact on properties, based on technically and scientifically based proposals (Mosavi Ozturk, & Chau, 2018), therefore, the prediction and modeling of discharges in a river basin is of utmost importance since it allows the issuance of warnings for flood control and management (Jimeno, Senent, Pérez, Pulido, & Cecilia, 2017). Likewise, by having a better accuracy of the records, it is possible to estimate and optimize the design of hydraulic works that help to counteract events such as El Niño (a), preserving hydraulic infrastructure of all types, housing, crop areas, animals, and most importantly, avoiding the loss of human lives (Cromwell *et al.*, 2021; Rodríguez, Díaz, Ballesteros, Rohrer, & Stoffel, 2019).

On the other hand, hydrological models such as distributed or semi-distributed ones require different types of data and inputs, while the application of machine and deep learning techniques simplifies the amount of these inputs (Basagaoglu, Chakraborty, & Winterle, 2021; Wang *et al.*, 2019). Thus, for example, the advantage of hydrological models that work with automatic and deep learning techniques, with respect to clustered or distributed models, is that they are able to simulate areas of different sizes (Sattari, Apaydin, Band, Mosavi, & Prasad, 2021;

Valderrama-Purizaca, Chávez-Barturen, Muñoz-Pérez, Tuesta-Monteza, & Mejía-Cabrera, 2021).

In that context, rainfall-runoff modeling plays a crucial role at several points in the management of hydrological resources, helping significantly to solve problems related to flood control and agricultural land protection (Fu *et al.*, 2020). The simulation of hydrometeorological variables is a very important process in hydrology, usually providing support for different water resources planning and management activities (Kim *et al.*, 2021). Designing a physical model for such phenomena is often costly and also requires absolute domain expertise, however, machine and deep learning techniques allow generating less costly, less complex and more efficient models (Qin, Liang, Chen, Lei, & Kang, 2019; Zhang *et al.*, 2020; Shi *et al.*, 2020). It is also worth mentioning that another of the hydrological processes that have been treated with automatic and deep learning techniques is the estimation of changes in river flow (Farfán, Palacios, Ulloa, & Avilés, 2020), lake water level prediction (Yaseen *et al.*, 2020) and evapotranspiration estimation (Afzaal, Farooque, Abbas, Acharya, & Esau, 2020); complex problems in the field of hydrology due to the incorporation of various hydrometeorological and morphological characteristics. Following that line, the objective of the present research was to apply Artificial Neural Networks to rainfall-runoff modeling and evaluate its model performance by means of the Nash-Sutcliffe efficiency metric (NSE), in a watershed of northern Peru, namely, the Chancay Lambayeque river basin.

Materials and methods

Methodology

The procedure developed in this study is presented in Figure 1. The data used in this study were from 12 hydrometeorological stations; 11 meteorological and 01 hydrological, from which 30 years of precipitation and flow records were considered on a daily scale respectively (normal hydrological), from 01/01/1991 to 31/12/2020, selected according to the availability of the information. With 80 % (24 years) of the data, the Long and Short Term Memory Model (LSTM) was trained and calibrated, and with 20 % it was validated (06 years). Along these lines, the Python v.2.7.8 programming language, libraries such as Keras v.2.4.4 and Tensor Flow v.2.3.2 were used to develop the ANNs. Likewise, libraries such as Numpy v.1.19.5 were used for numerical processing, Pandas v.1.1.4 to generate tables, Matplotlib v.3.3.3 to plot the data and Seaborn to correlate data. Scikit-Learn v.0.23.4 was used for the implementation of Adam's optimization and Mean Squared Error. The code was run on a laptop with Intel(R) Core (TM) i5-4210U 2.40GHz x 4 64-bit CPU and 12.0GB RAM in Windows 10 environment.

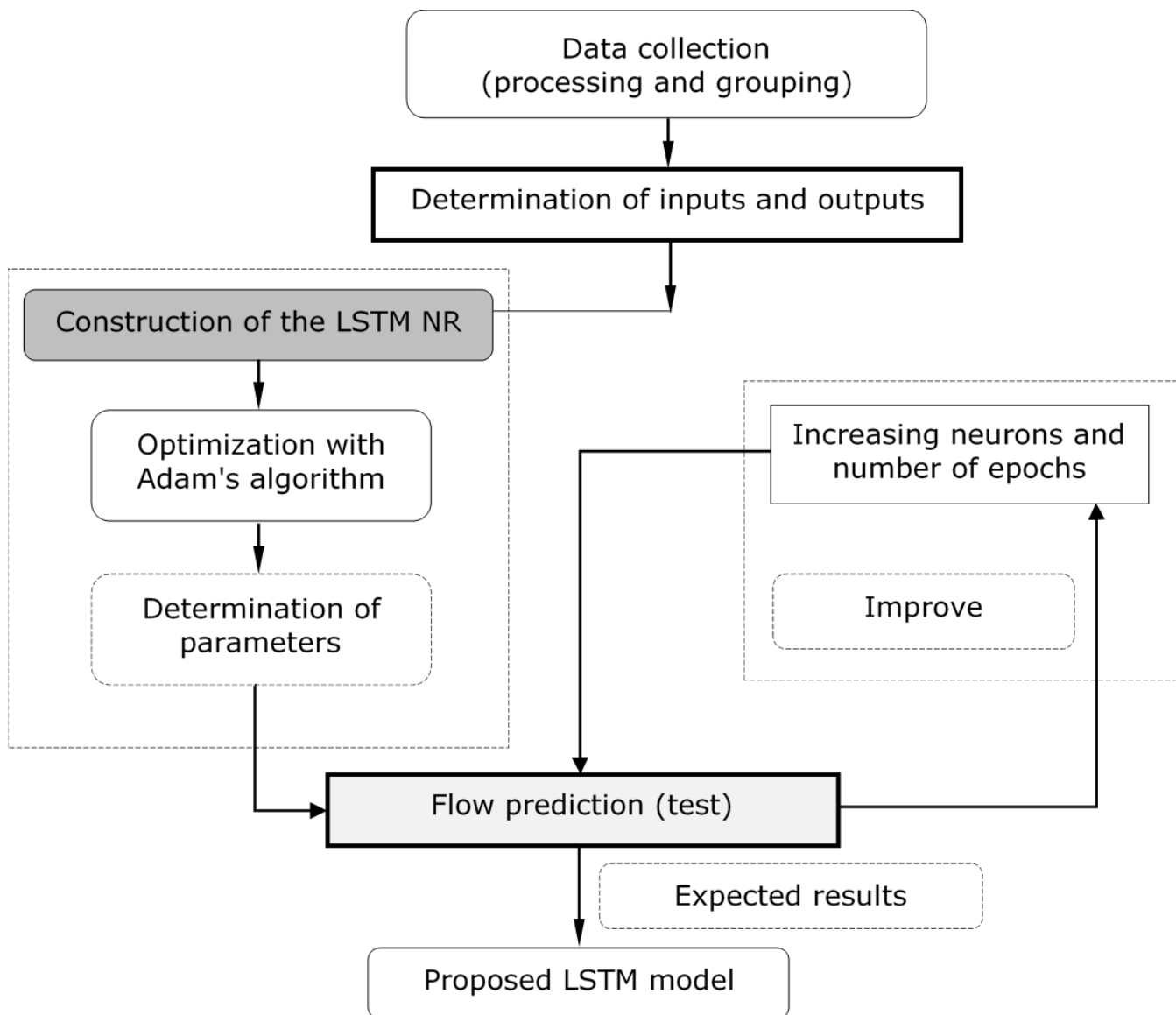


Figure 1. Procedure developed in rainfall-runoff modeling.

Study area

The study area is the Chancay Lambayeque river basin, located in northern Peru, which from time to time is exposed to events such as floods, the occurrence of the El Niño phenomenon, which in turn generates heavy rainfall, flooding of streams, high temperatures, among others. Along these lines, rainfall-runoff modeling in the Chancay Lambayeque River basin by applying Neural Network models is a fundamental tool for generating hydro-meteorological information at strategic points. On the other hand, according to the geomorphological parameters of the basin, Figure 2 shows an area of 4043.73 km² (large basin), a perimeter of 432.86 km and a hypsometric curve in the Maturity phase. Likewise, it is an elongated basin with a slow response as maximum runoff to precipitation, with a drainage density of Medium.



Figure 2. Chancay River Basin and hydrometeorological stations considered in the study.

Artificial Neural Networks (ANNs)

ANNs are mathematical modeling systems that perform parallel data processing (Adamowski, Chan, Prasher, Ozga-Zielinski, & Sliusarieva, 2012). The first ANN was developed in 1943 by McCulloch and Pitts with the purpose of imitating the functioning of biological neural networks by interconnecting them, as shown in Figure 3. In hydrology, they began to be used in the 1990s, as was done by French, Krajewski and Cuykendall

(1992) for rainfall prediction (Choubin, Khalighi-Sigaroodi, Malekian, & Kişi, 2014). Along these lines, some researches recommend ANNs as one of the most suitable modeling techniques that provide an acceptable generalization capacity and speed compared to other conventional models (Nabipour, Dehghani, Shamsirband, & Mosavi, 2020).

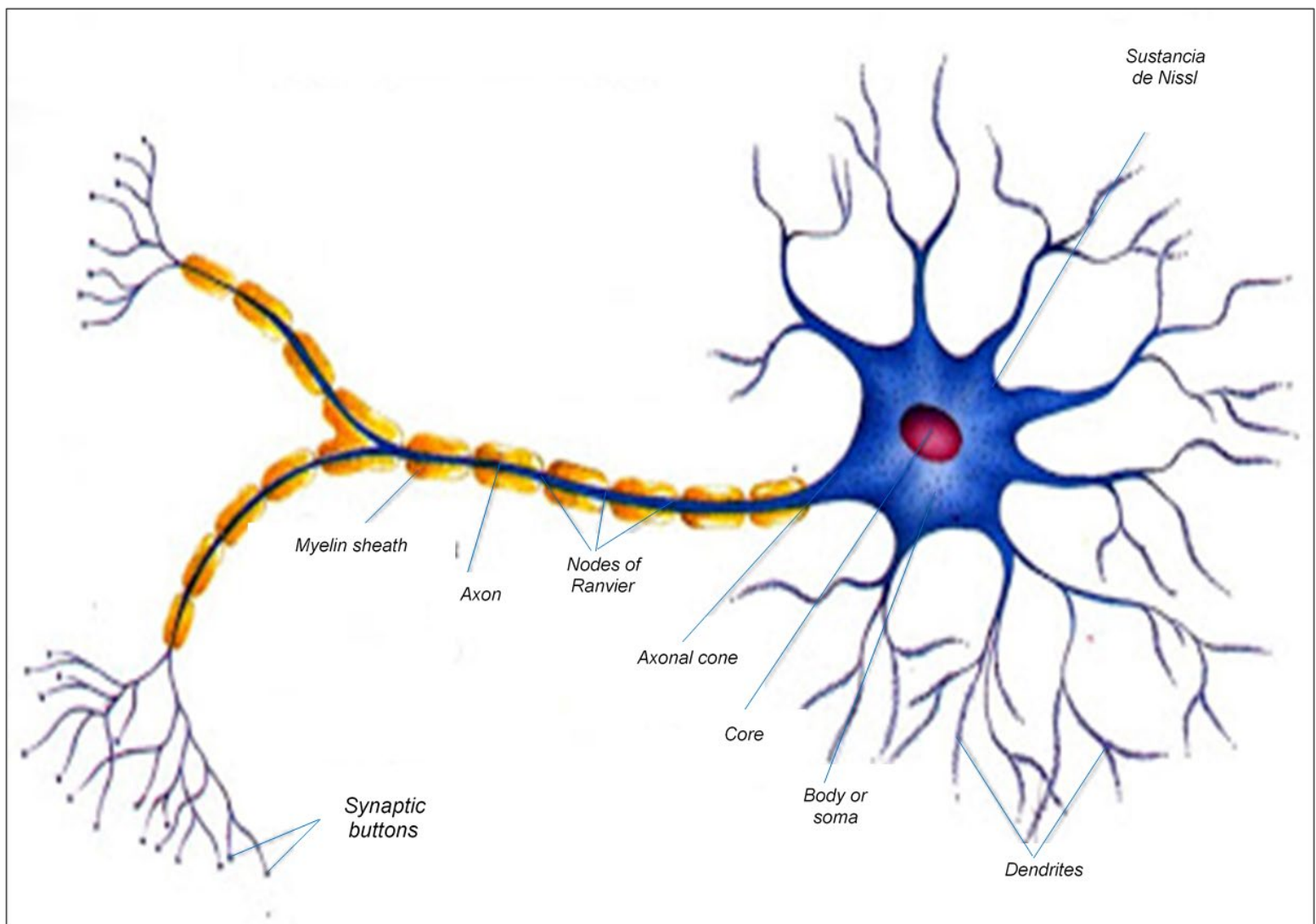


Figure 3. General structure of a biological neuron.

An artificial neural network generally consists of one to several layers, one input, some hidden and one output (Béjar, Valeriano, Ilachoque, & Sulla, 2016), as shown in Figure 4, where each layer has several neurons that receive inputs from the previous or external layer and convert them into an output or input signal to be used by other neurons in the next layer (Laqui-Vilca, 2010).

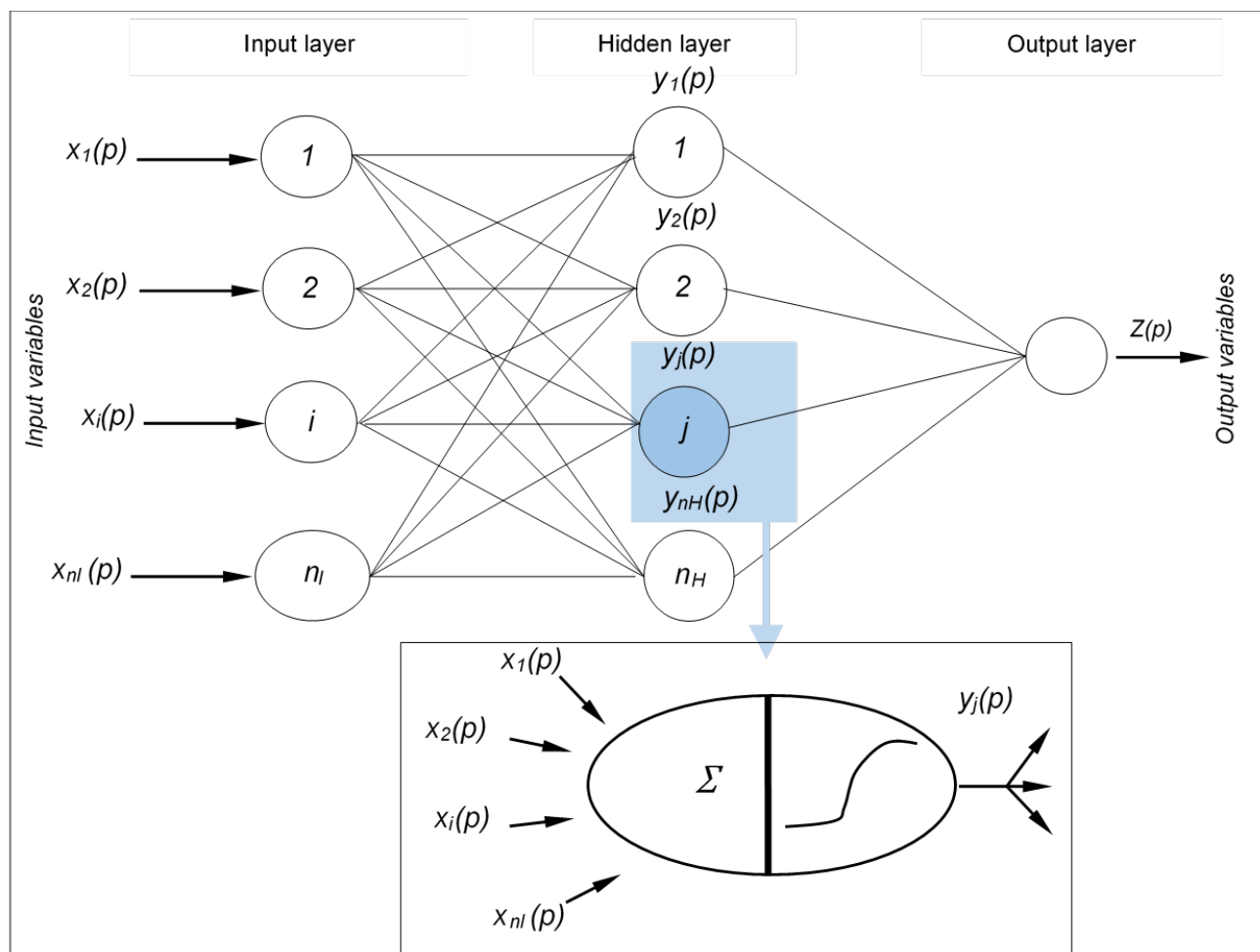


Figure 4. Three-layer ANN architecture (input, hidden and output) (Hu *et al.*, 2018).

Neural network training

It is the learning process by which configurations are made to the NR, so that the desired outputs are produced according to the inputs (Rezaeianzadeh, Tabari, & Yazdi, 2014). As part of the training, the weights w_{ij} are adjusted and should converge to consistent values.

Twelve variables were selected for training the neural network. The first variable represents the hydrological flow corresponding to the Racarrumi River measured in m^3/s . The next 11 variables represent the rainfall in 11 localities that generate tributaries to the Racarrumi River measured in mm. The data set was constructed based on measurements obtained from Senhami's hydrological and hydrometric stations located in the localities near the Racarrumi River. Likewise, 10 958 records corresponding to 30 years of flow and rainfall information were compiled and treated, with maximum and minimum values as shown in Table 1. The dataset incorporated numerical values.

Table 1. Maximum and minimum values at hydrological and hydrometric stations according to locality name.

Nº	Station	Type	Maximun	Minimum
01	Racarrumi	Estación Hidrológica	358.541 m ³ /s	0.88 m ³ /s
02	Chugur	Estación hidrométrica	207.73 mm	0 mm
03	Udima		230 mm	
04	Llama		203.08 mm	
05	Chancay		248.95 mm	
06	Tocmoche		110 mm	
07	Espinal		107 mm	
08	Puchaca		150.20 mm	
09	Cayalti		77.30 mm	
10	Jayanca		120.80 mm	
11	Lambayeque		71.30 mm	
12	Reque		60.40 mm	

Based on previous works (Hu *et al.*, 2018; Fan *et al.*, 2020; Fu *et al.*, 2020) the Long-Short Term Memory type Neural networks known as (LSTM) network is special kind of recurrent neural network (RNN) structure, overcoming the weakness of the traditional RNN to learn long-term dependencies. The deep in time structure of LSTM enables it to lean when to forget and how long to retain the state information through the specially designed gates and memory cells as shown in Figure 5.

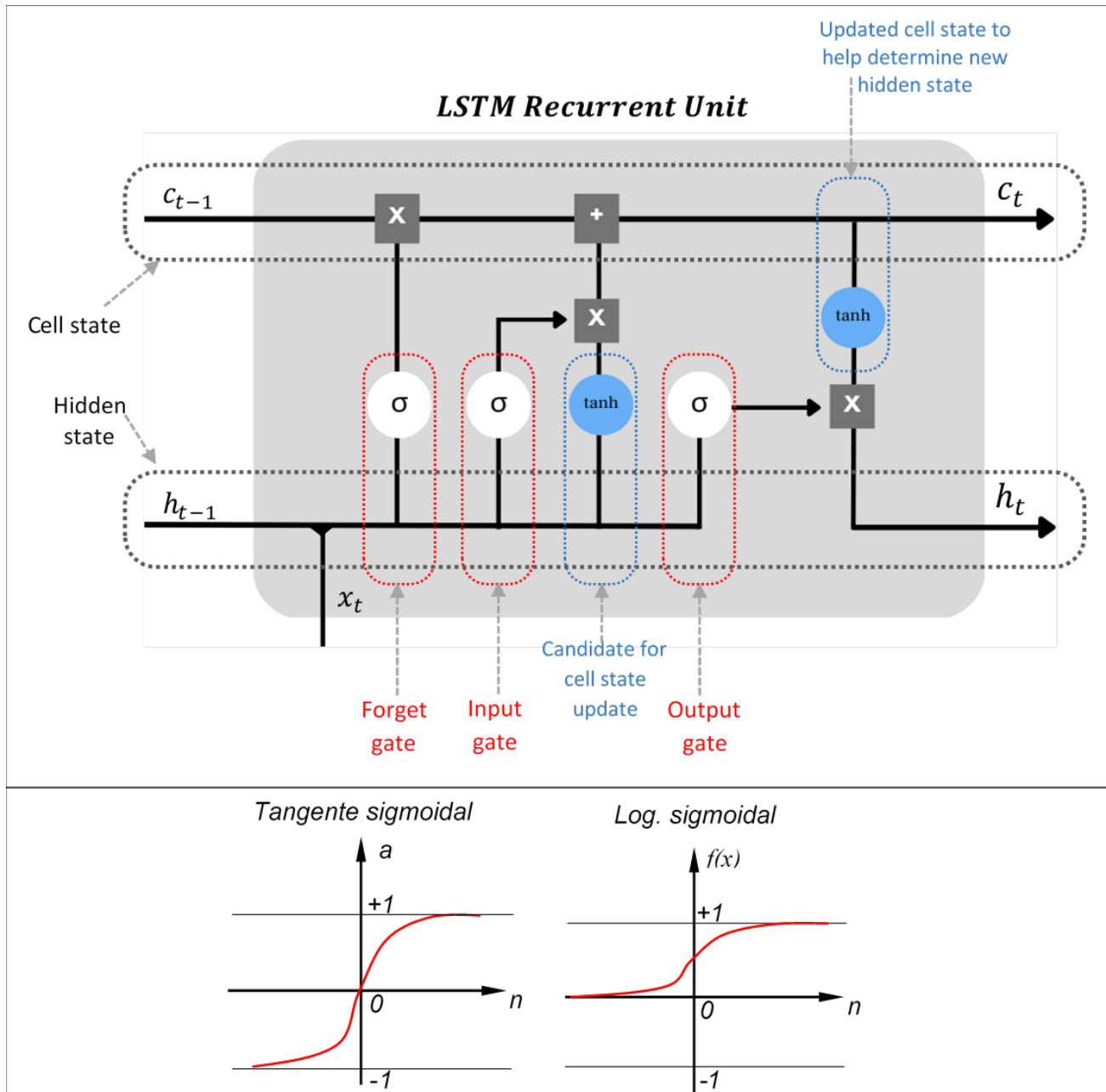


Figure 5. Learning process and neural network transfer functions (Yaseen *et al.*, 2020).

LSTMs were used in this research because they allow modeling the time dependence of water flow in the Racarrumi River. This is important because the river flow can vary greatly over 30 years due to factors such as rainfall, evaporation, and other climatic factors. The ability of LSTM networks to retain long-term information allows them to model these temporal dependencies and make accurate predictions of river flow, allowing a return time (RT) of up to 1 000 years.

In addition, LSTM networks can learn from complex patterns in the data, which means that they can identify nonlinear relationships between variables that affect river flow, such as rainfall at the 11 locations that are tributaries to the Racarrumi River.

In order to estimate the flow rate of the Racarrumi River, an output variable was established in the LSTM Neural Network, whose value represents the amount of flow in m^3/s . The architecture of the network was defined with an input layer of 12 neurons connected to the input vector, 4 hidden layers with 50 LSTM neurons each and a dense layer with a single neuron based on the RELU function, as shown in Figure 6.

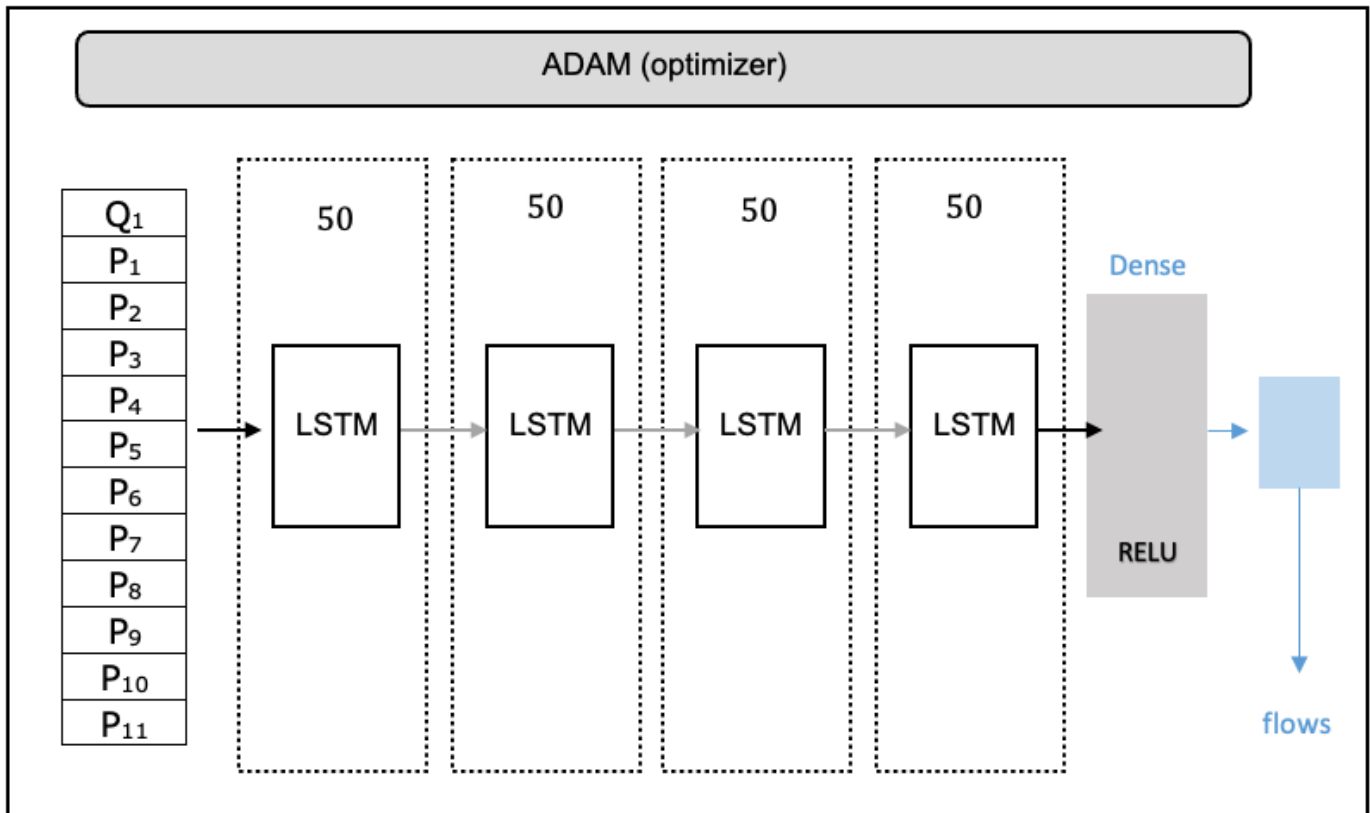


Figure 6. Neural Network Architecture based on LSTM.

The constructed model uses LSTM elements and a dense layer and the Adaptive Moment Estimation (ADAM) optimization algorithm was also selected because it is designed to autonomously adapt to the gradient characteristics of the loss function being optimized. This means that ADAM automatically adjusts the learning rate during training to adapt to different conditions of the flow and precipitation data to avoid the problems of falling into local minima of the loss function. Likewise, ADAM uses a first- and second-order moving average of the gradients, which allows it to efficiently handle the noise in the data during the Racarrumi

River flow estimation. The mathematical form of ADAM is shown in the following equation:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Where:

θ_t = vector of weights at time t during training

α = learning rate

$\hat{m}_t$ = estimate of the first moment of the gradient at time t

$\hat{v}_t$ = estimate of the second moment of the gradient at time t

ϵ = smoothing term to avoid division by zero

Results and discussion

Daily scale hydrometeorological information of the Chancay Lambayeque river basin

Hydrometeorological information was collected from 01/01/1991 to 12/31/2020, that is, 30 years of records, for a total of 10958 daily data for each station. Table 2 shows the 11 meteorological stations and one hydrological station (No. 12), distributed throughout the basin, as well as in the surrounding areas, which are expected to be part of the Pacific slope, and in particular the Zonal Directorate No. 02 of SENAMHI, from

which precipitation records were considered, the same that were downloaded from its database. web: <https://www.senamhi.gob.pe/?p=descarga-datos-hidrometeorologicos>.

Table 2. Stations used in rainfall-runoff modeling.

N°	Station	Basin	Department	Elevation (msnm)	Geographic Coordinates	
					Latitude (S)	Longitude (O)
01	Chugur	Chancay Lambayeque	Cajamarca	2 742	6° 40' 10.02"	78°44'17.06"
02	Udima	Zaña	Cajamarca	2 466	6°48'53.08"	79°5'37.56"
03	Llama	Chancay Lambayeque	Cajamarca	2 096	6°30'51.95"	79°7'21.43"
04	Chancay Baños	Chancay Lambayeque	Cajamarca	1 639	6°34'29.61"	78°52'1.96"
05	Tocmoche	La Leche	Cajamarca	1 399	6°24'36.33"	79°21'20.58"
06	El espinal	Zaña	Lambayeque	409	6°49'3.1"	79°12'5.8"
07	Puchaca	La Leche	Lambayeque	336	6°22'25"	79°28'10.25"
08	Cayaltí	Zaña	Lambayeque	90	6°52'50.86"	79°32'49.25"
09	Jayanca	Motupe	Lambayeque	78	6°19'53.73"	79°46'7.29"
10	Lambayeque	Chancay Lambayeque	Lambayeque	18	6°44'3.75"	79°54'35.4"
11	Reque	Chancay Lambayeque	Lambayeque	13	6°53'10.07"	79°50'7.8"
12	Racarrumi	Chancay Lambayeque	Lambayeque	254	6°37'59.68"	79°18'35.14"

Likewise, hydrometric information from the "Racarrumi" station was used, data available at: http://www.peot.gob.pe/tinajones/busca_tin_inf.php

The type of neural networks used were the recurrent ones, the same that are usually used for time series prediction (Abbot & Marohasy, 2014), in this sense, the modeling was divided into five stages.

Stage 01: Data pre-processing

As part of this stage, Figure 7 details the import of libraries; Numpy, with which the calculation and numerical analysis of the data was performed.

```
# Import of libraries.
import numpy as np           #Library for numerical processing.
import pandas as pd         #Library to generate tables.
import matplotlib.pyplot as plt #Library for plotting.
import seaborn as sns       #Data correlation library.
from google.colab import drive #Library to import data from Google Drive
drive.mount("/content/drive")
```

Figure 7. Libraries used for modeling in Python.

The Pandas library was used to filter the columns to be used from the tables, visualizing the records of the hydrometeorological stations shown in Figure 8.

```
# Import the data set (dataset).
dataset = pd.read_excel("/content/drive/MyDrive/ChancayL.xlsx", sheet_name="ChancayL_CcP_19912020", index_col=0)
dataset
```

	Racarrumi	Chugur	Udima	Llama	ChancayB	Tocmoche	Espinal	Puchaca	Cayalti	Jayanca	Lambayeque	Reque
Fecha												
1991-01-01	33.270	0.3	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0
1991-01-02	26.470	2.0	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0
1991-01-03	21.960	3.3	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0
1991-01-04	19.410	0.0	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0
1991-01-05	18.370	0.0	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
2020-12-27	62.801	0.0	0.0	0.000000	0.0	0.0	0.89044	0.278716	0.0	0.0	0.0	0.0
2020-12-28	57.460	0.0	0.0	0.000000	0.0	0.0	0.89044	0.278716	0.0	0.0	0.0	0.0
2020-12-29	48.770	0.0	0.0	0.000000	0.0	0.0	0.89044	0.278716	0.0	0.0	0.0	0.0
2020-12-30	41.386	0.0	0.0	0.000000	0.0	0.0	0.89044	0.278716	0.0	0.0	0.0	0.0
2020-12-31	34.818	10.6	0.0	10.507946	0.0	0.0	0.89044	0.278716	0.0	0.0	0.0	0.0

10958 rows x 12 columns

Figure 8. Pandas library application.

With the Matplotlib library, the graphs were created as shown in Figure 9.

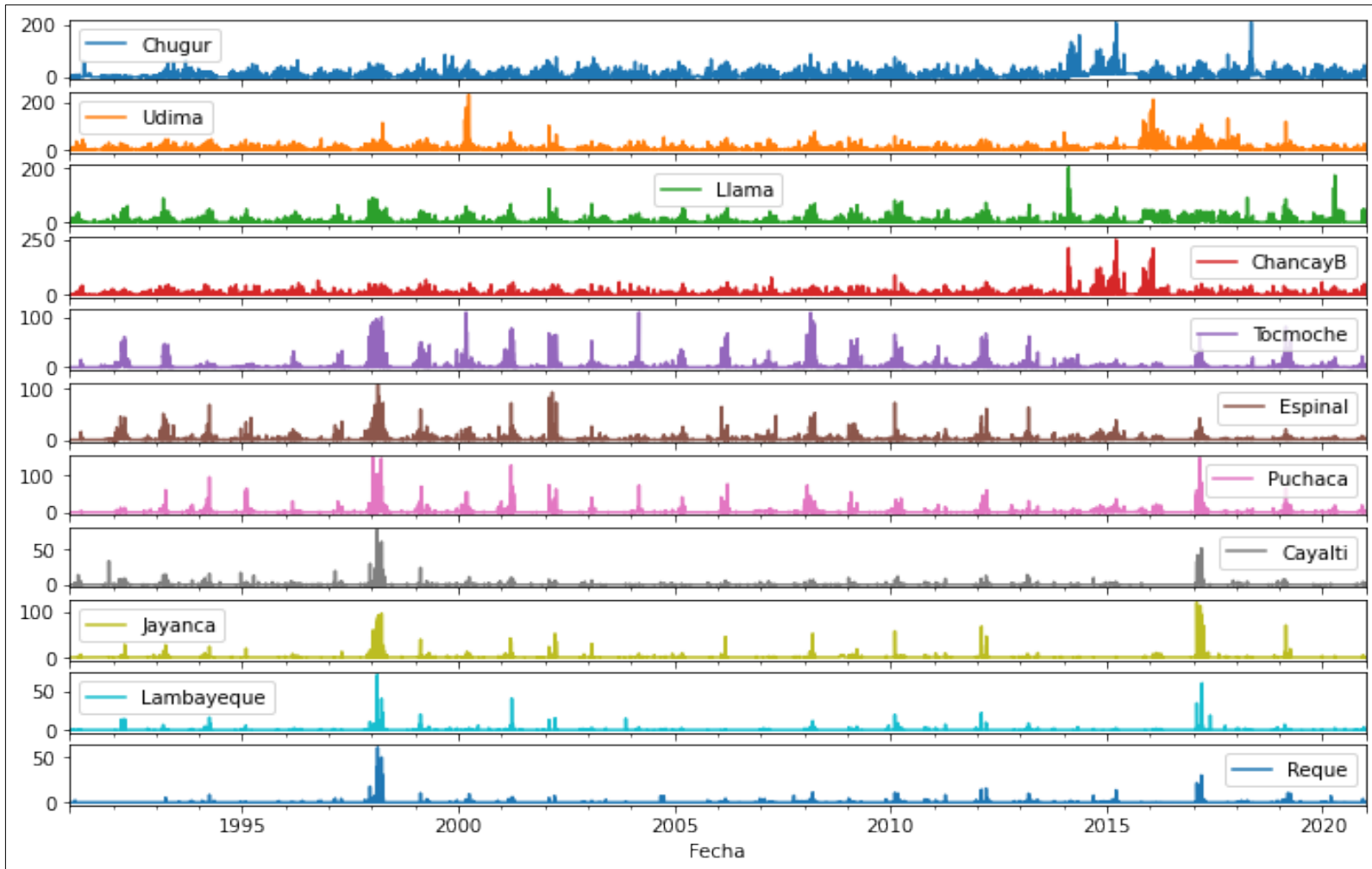


Figure 9. Precipitation histograms plotted with the Matplotlib library.

Finally, Seaborn was used to perform the numerical correlations between hydrometeorological stations as shown in Figure 10.

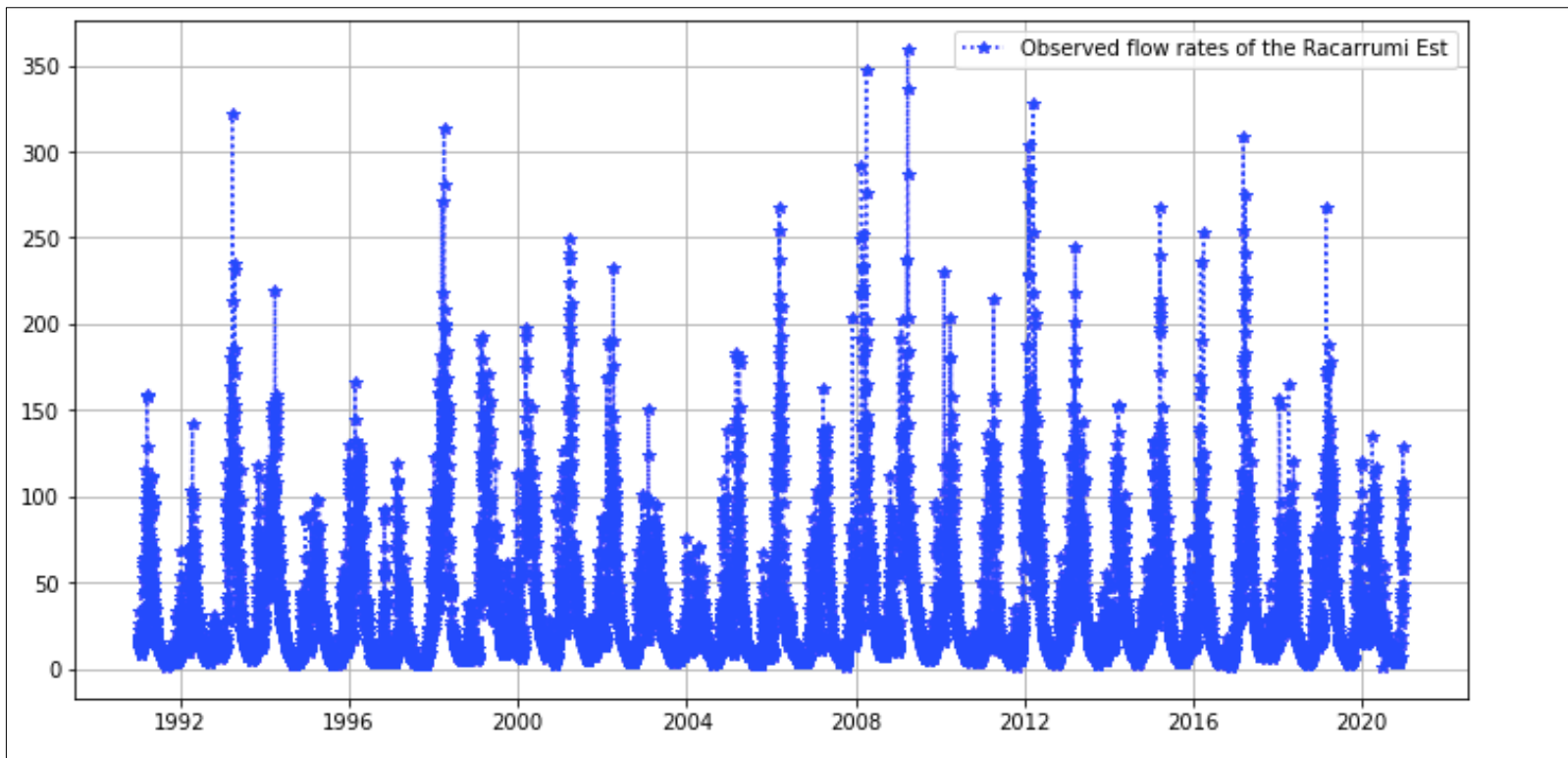


Figure 10. Hydrograph of flow rates recorded at "Racarrumi" station, period 01/01/1991 - 12/31/2020; plotted with the Matplotlib library in Python.

Figure 11 and Figure 12 show the correlation between the stations, showing that those that are close to each other, or are at a similar elevation, have values close to 1.0, as is the case of the stations Chugur and Chancay Baños (nearby stations), or the stations Lambayeque and Reque (similar elevation).

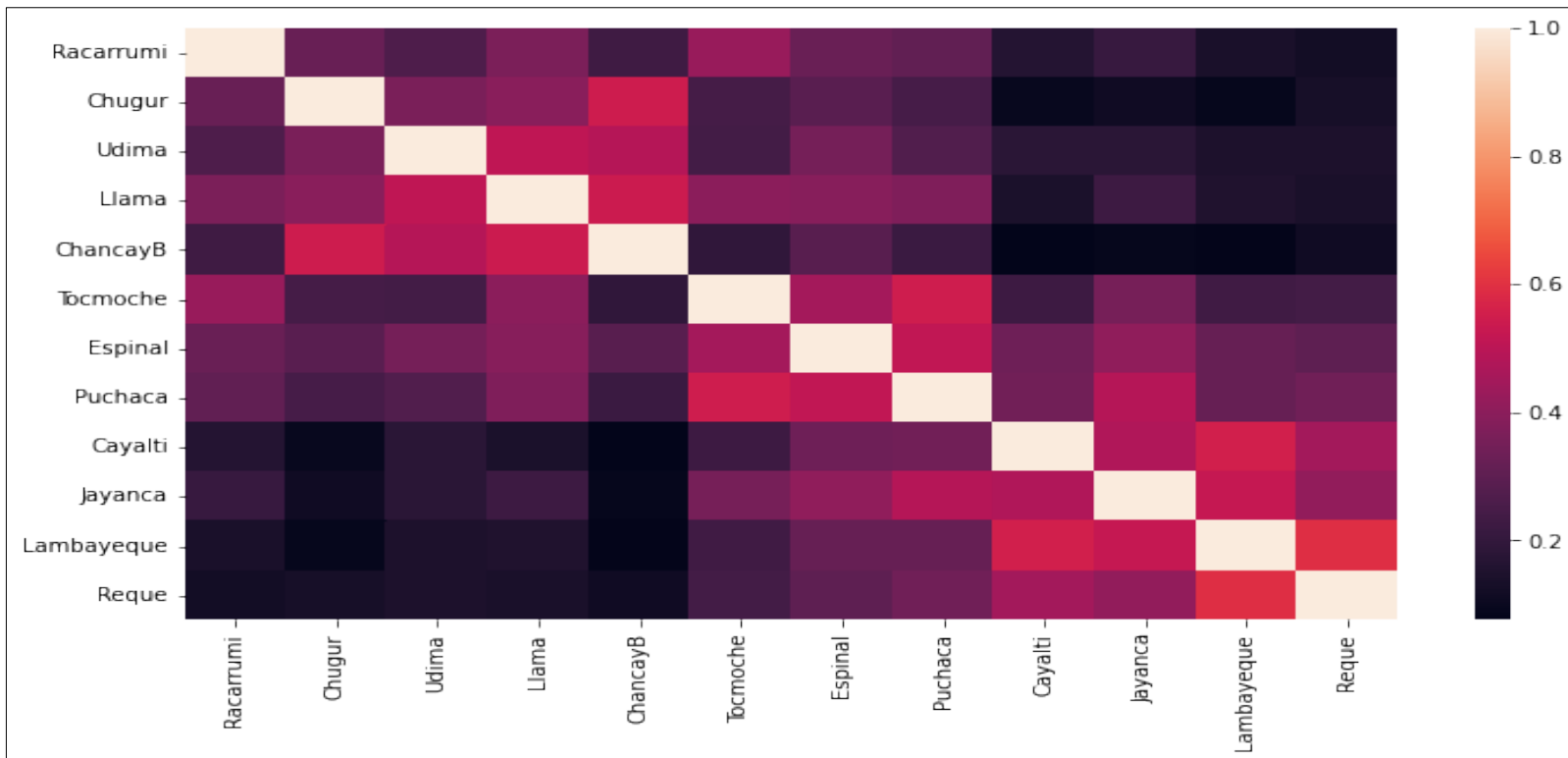


Figure 11. Correlation diagram generated with the Seaborn library and the Matplotlib library in Python.

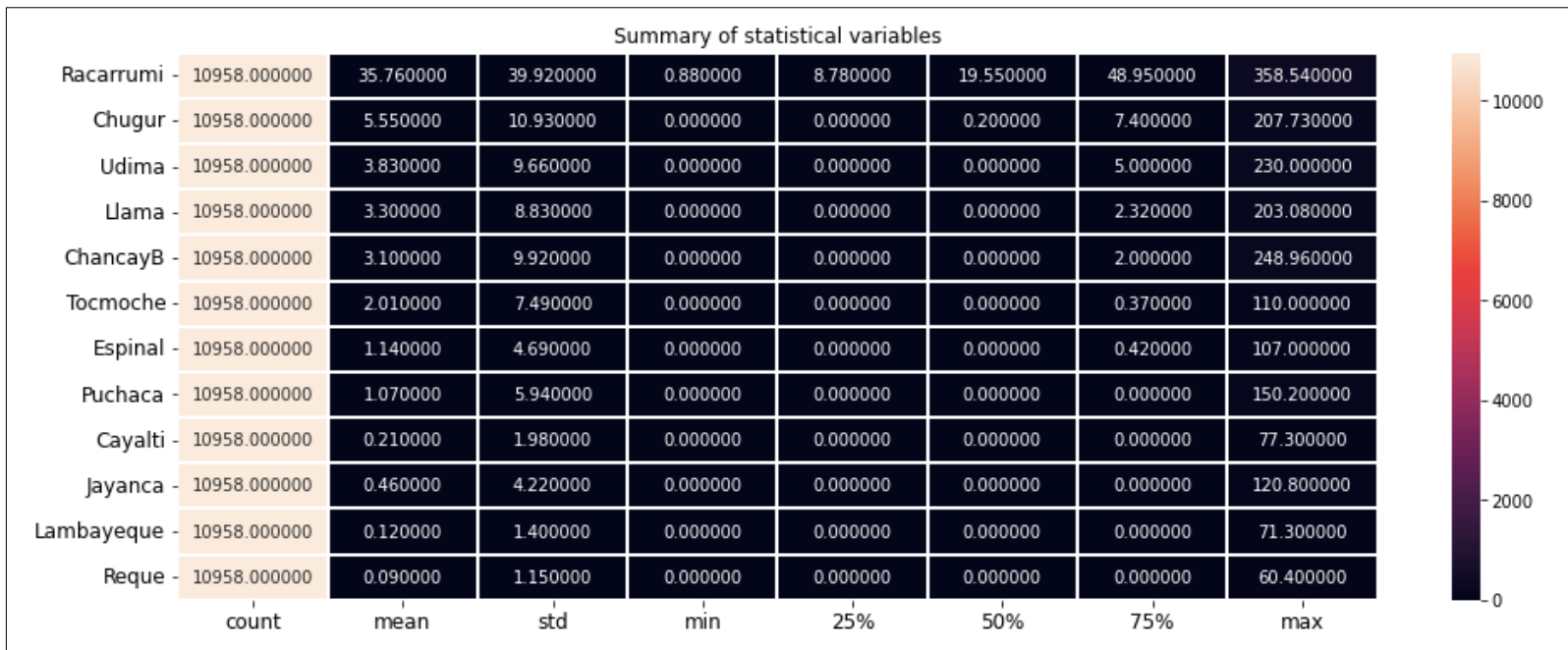


Figure 12. Statistical summary of hydrometeorological records generated with the Seaborn library and the Matplotlib library in Python.

Figure 13 below shows the code used to define the data for model training (80 % of the information), *i.e.*, the station records from 01/01/1991 to 12/31/2014.

```
# The training dataset is imported.
dataset_train = pd.read_excel("/content/drive/MyDrive/ChancayL.xlsx", sheet_name="ChancayL_CCP19912014")
training_set = dataset_train.iloc[:, 1:13].values
training_set

array([[3.32700000e+01, 3.00000000e-01, 0.00000000e+00, ...,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
       [2.64700000e+01, 2.00000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
       [2.19600000e+01, 3.30000000e+00, 0.00000000e+00, ...,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
       ...,
       [6.20670000e+01, 1.32560844e+01, 9.17965126e+00, ...,
        2.12191045e-01, 1.07557714e-01, 4.00000000e-01],
       [5.28580000e+01, 1.80588760e+01, 8.40026379e+00, ...,
        1.65088375e-01, 3.20379846e-02, 0.00000000e+00],
       [4.15230000e+01, 1.32560844e+01, 9.17965126e+00, ...,
        1.65088375e-01, 3.20379846e-02, 0.00000000e+00]])
```

Figure 13. Code for the import of hydrometeorological records used in model training.

Subsequently, Figure 14 shows the scaling of the training data with the "MinMaxScaler" function, so these values ranged between 0 and 1.

```
# Scale the features.
from sklearn.preprocessing import MinMaxScaler

# The scaler is saved so that the "fit" function can be used later.
sc = MinMaxScaler(feature_range = (0, 1))

# The "fit" function is applied to the scaling performed and the values are transformed.
training_set_scaled = sc.fit_transform(training_set)
training_set_scaled

array([[0.09056061, 0.0018826 , 0.          , ..., 0.          , 0.          ,
        0.          ],
       [0.0715482 , 0.01255066, 0.          , ..., 0.          , 0.          ,
        0.          ],
       [0.05893849, 0.02070858, 0.          , ..., 0.          , 0.          ,
        0.          ],
       ...,
       [0.1710754 , 0.08318629, 0.03991153, ..., 0.00220344, 0.00150852,
        0.00662252],
       [0.14532756, 0.11332538, 0.03652289, ..., 0.00171431, 0.00044934,
        0.          ],
       [0.11363554, 0.08318629, 0.03991153, ..., 0.00171431, 0.00044934,
        0.          ]])
```

Figure 14. Transformation of values of hydrometeorological records used in model training.

Figure 15 details the creation of a data structure with 60 time steps, i.e., every 60 days, and with 12 rows representing the number of hydrometeorological stations, thus creating "arrays" for each of the station data used for the prediction. It is worth mentioning that only one matrix was defined for the case of flow rates, since only one flow station was considered in this study, which was called "and train".

```
# We create a data structure with 60 timesteps and 12 outputs of "n" rows with 60 days length for each variable.
# An array is created for each variable to be used for prediction.
Racarrumi_train = []
Chugur_train = []
Udima_train = []
Llama_train = []
ChancayB_train = []
Tocmoche_train = []
Espinal_train = []
Puchaca_train = []
Cayalti_train = []
Jayanca_train = []
Lambayeque_train = []
Reque_train = []
```

Figure 15. Code designed for the creation of the matrices containing the hydrometeorological records used in the training of the model.

Then with the Numpy library the data was resized, so a dimension was added to each variable shown in Figure 16 to be the size of (8 706, 60, 1); *i.e.* 8 706 rows and 60 columns.

```
# The data is resized. We also add a dimension to each variable so that they are of size (8706, 60, 1).
Racarrumi_train_resized = np.reshape(Racarrumi_train, (Racarrumi_train.shape[0], Racarrumi_train.shape[1], 1))
Chugur_train_resized = np.reshape(Chugur_train, (Chugur_train.shape[0], Chugur_train.shape[1], 1))
Udima_train_resized = np.reshape(Udima_train, (Udima_train.shape[0], Udima_train.shape[1], 1))
Llama_train_resized = np.reshape(Llama_train, (Llama_train.shape[0], Llama_train.shape[1], 1))
ChancayB_train_resized = np.reshape(ChancayB_train, (ChancayB_train.shape[0], ChancayB_train.shape[1], 1))
Tocmoche_train_resized = np.reshape(Tocmoche_train, (Tocmoche_train.shape[0], Tocmoche_train.shape[1], 1))
Espinal_train_resized = np.reshape(Espinal_train, (Espinal_train.shape[0], Espinal_train.shape[1], 1))
Puchaca_train_resized = np.reshape(Puchaca_train, (Puchaca_train.shape[0], Puchaca_train.shape[1], 1))
Cayalti_train_resized = np.reshape(Cayalti_train, (Cayalti_train.shape[0], Cayalti_train.shape[1], 1))
Jayanca_train_resized = np.reshape(Jayanca_train, (Jayanca_train.shape[0], Jayanca_train.shape[1], 1))
Lambayeque_train_resized = np.reshape(Lambayeque_train, (Lambayeque_train.shape[0], Lambayeque_train.shape[1], 1))
Reque_train_resized = np.reshape(Reque_train, (Reque_train.shape[0], Reque_train.shape[1], 1))
```

Figure 16. Code designed for the resizing of the data used in the training of the model.

Finally, stage 01 of the data pre-processing ends with the creation of a single matrix of size (8 706, 60, 12) shown in Figure 17, where 8 706 is the number of rows, 60 the number of columns in which each data set was grouped, and 12 the number of stations; 11 pluviometric and 01 hydrometric (Racarrumi).

```
# The array is created which will result in an array of the size of (8706, 60, 12).
x_train = np.append(Racarrumi_train_reshaped, (Chugur_train_reshaped), axis = 2)
x_train = np.append(x_train, (Udima_train_reshaped), axis = 2)
x_train = np.append(x_train, (Llama_train_reshaped), axis = 2)
x_train = np.append(x_train, (ChancayB_train_reshaped), axis = 2)
x_train = np.append(x_train, (Tocmoche_train_reshaped), axis = 2)
x_train = np.append(x_train, (Espinal_train_reshaped), axis = 2)
x_train = np.append(x_train, (Puchaca_train_reshaped), axis = 2)
x_train = np.append(x_train, (Cayalti_train_reshaped), axis = 2)
x_train = np.append(x_train, (Jayanca_train_reshaped), axis = 2)
x_train = np.append(x_train, (Lambayeque_train_reshaped), axis = 2)
x_train = np.append(x_train, (Reque_train_reshaped), axis = 2)
x_train
```

Figure 17. Code designed for the resizing of the data used in the training of the model.

Stage 02: Construction of the Recurrent Neural Network (RNN)

As part of this stage, the Keras library was imported with which the neural network modeling was carried out, and the Dense function was used to define the number of layers and neurons in the model, the LSTM function

was used to define the type of neural networks, and the Dropout function was used to define the output layer.

Figure 18 shows the neural network created by 04 layers, where the first, second, third and fourth were made up of 50 neurons each; while the output layer consisted of 01 neuron that represented the output flows.

```
# The first LSTM layer and the regularization by "Dropout" are added.
model.add(LSTM(units = 50, return_sequences = True, input_shape = (x_train.shape[1], x_train.shape[2])))
model.add(Dropout(0.2))

# The second LSTM layer and the regularization by "Dropout" are added.
model.add(LSTM(units = 50, return_sequences = True))
model.add(Dropout(0.2))

# The third LSTM layer and the regularization by "Dropout" are added.
model.add(LSTM(units = 50, return_sequences = True ))
model.add(Dropout(0.2))

# The fourth LSTM layer and the regularization by "Dropout" are added.
model.add(LSTM(units = 50))
model.add(Dropout(0.2))

# the output layer is placed.
model.add(Dense(units = 1))

# Execution of the Recurrent Neural Network (RNN).
model.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

Figure 18. Layer and neuron encoding of the NN.

The "Adam" model was used as optimizer and the "mean square error" was used for the losses. Figure 19 shows that 32 epochs were

defined for training the neural networks, i.e. 32 simultaneous runs, while 80 % of the information (24 years of daily records) was taken as data.

```
# The RNN is adjusted to the training set, it should be noted that 80% of the 30 years of records were taken for
model.fit(x_train, y_train, epochs = 100, batch_size = 32)
```

```
Epoch 1/100
273/273 [=====] - 40s 116ms/step - loss: 0.0051
Epoch 2/100
273/273 [=====] - 35s 128ms/step - loss: 0.0026
Epoch 3/100
273/273 [=====] - 33s 122ms/step - loss: 0.0018
Epoch 4/100
273/273 [=====] - 31s 115ms/step - loss: 0.0015
Epoch 5/100
273/273 [=====] - 33s 122ms/step - loss: 0.0014
Epoch 6/100
273/273 [=====] - 31s 115ms/step - loss: 0.0012
Epoch 7/100
273/273 [=====] - 35s 128ms/step - loss: 0.0012
Epoch 8/100
273/273 [=====] - 33s 122ms/step - loss: 0.0011
Epoch 9/100
273/273 [=====] - 31s 115ms/step - loss: 0.0011
Epoch 10/100
273/273 [=====] - 33s 121ms/step - loss: 0.0010
Epoch 11/100
273/273 [=====] - 31s 114ms/step - loss: 0.0010
Epoch 12/100
273/273 [=====] - 33s 122ms/step - loss: 0.0011
```

Figure 19. Neural network model training.

Stage 03: Adjust predictions and visualize results

As part of this stage, Figure 20 shows the import of the observed flows considered for the model validation stage, i.e. 06 years of daily records, namely from 01/01/2015 to 12/31/2020 (20 % of the total information).

```
# The flow rates observed at the "Racarrumi" hydrometric station from 01/01/2015 to 31/12/2020 (2192 data)
dataset_test = pd.read_excel("/content/drive/MyDrive/ChancayL.xlsx", sheet_name="Racarrumi20152020")
caudales_racarrumi_20152020 = dataset_test.iloc[:, 1:2].values
caudales_racarrumi_20152020

array([[40.821],
       [38.508],
       [39.18 ],
       ...,
       [48.77 ],
       [41.386],
       [34.818]])
```

Figure 20. Import of observed flow rates for model validation.

Taking the precipitation and flow data, the prediction of these was carried out, considering the last 06 years of record. For this purpose, as in the training stage, the flow rates were scaled between 00 and 01 as shown in Figure 21.

```
# Flow prediction is performed with the RNR from 01/01/2015 to 12/31/2020 (2192 data).
dataset_total = pd.concat((dataset_train[ ['Racarrumi', 'Chugur', 'Udima', 'Llama', 'ChancayB', 'Tocmoche', 'Espinal'
inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values

# The scaler is used for the data set to which the fit was applied to generate the training data set "x_train".
inputs = sc.transform(inputs)
inputs

array([[0.02300223, 0.08318629, 0.03991153, ..., 0.00171431, 0.00044934,
        0.],
       [0.02744498, 0.08318629, 0.03991152, ..., 0.00171431, 0.00044934,
        0.],
       [0.03443484, 0.08318629, 0.03991152, ..., 0.00220344, 0.00150852,
        0.00662252],
       ...,
       [0.13389774, 0., ..., 0., ..., 0., ..., 0.],
       [0.11325249, 0., ..., 0., ..., 0., ..., 0.],
       [0.09488874, 0.06651848, ..., 0., ..., 0., ..., 0.]])
```

Figure 21. Scaling of the data considered in the validation.

In this sense, Figure 22 and Figure 23 shows the creation of the data set for the validation "x_test" with the variables of the training (test) data set and the lists were converted with the Numpy library to a single array, whose size was 2 192 rows by 60 columns containing the data from the 12 hydrometeorological stations.

```
# Creation of the data set for validation "x_test" with the variables of the training (test) data set.
# Creation of the data structure with 60 timesteps and 1 output.
Racarrumi_test = []
Chugur_test = []
Udima_test = []
Llama_test = []
ChancayB_test = []
Tocmoche_test = []
Espinal_test = []
Puchaca_test = []
Cayalti_test = []
Jayanca_test = []
Lambayeque_test = []
Reque_test = []
```

Figure 22. Creation of the data set for validation "x_test".

```
# An array of size (2192, 60, 12) is created.
x_test = np.append(Racarrumi_test_reshaped, (Chugur_test_reshaped), axis = 2)
x_test = np.append(x_test, (Udima_test_reshaped), axis = 2)
x_test = np.append(x_test, (Llama_test_reshaped), axis = 2)
x_test = np.append(x_test, (ChancayB_test_reshaped), axis = 2)
x_test = np.append(x_test, (Tocmoche_test_reshaped), axis = 2)
x_test = np.append(x_test, (Espinal_test_reshaped), axis = 2)
x_test = np.append(x_test, (Puchaca_test_reshaped), axis = 2)
x_test = np.append(x_test, (Cayalti_test_reshaped), axis = 2)
x_test = np.append(x_test, (Jayanca_test_reshaped), axis = 2)
x_test = np.append(x_test, (Lambayeque_test_reshaped), axis = 2)
x_test = np.append(x_test, (Reque_test_reshaped), axis = 2)
x_test
```

Figure 23. Code for the creation of a single array for model validation.

Figure 24 shows the flow prediction from 01/01/2015 to 12/31/2020, with those that were validated in the model and where later in Figure 25 we visualize the code with the Matplotlib library prior to the results shown in Figure 26 in hydrographs.

```
# Flow forecasting is performed from 01/01/2015 to 12/31/2020 (2192 data).
predicted_caudales_racarrumi_20152020 = model.predict(x_test)

69/69 [=====] - 4s 31ms/step

# Number of data
predicted_caudales_racarrumi_20152020.shape
```

Figure 24. Code for flow prediction with which the model was validated.

```
# Finally, the results are displayed.
plt.figure(figsize=(25,9)) #Window size.
plt.plot(caudales_racarrumi_20152020[:, 0], color = 'red', label = 'Observed flow rates of Racarrumi Est. from 01/0
plt.plot(Q_simulados, color = 'blue', label = 'Simulated flow rates of Racarrumi Est. from 01/01/2015 to 12/31/2020
plt.title("Prediction with an RNN of the Racarrumi Est. flows, year 2015 to 2020.")
plt.xlabel("Date (Years 2015-2020)")
plt.ylabel("Simulated and observed flow rates at Racarrumi Station")
plt.legend()
plt.show()
```

Figure 25. Coding with the Matplotlib library for visualization of results.

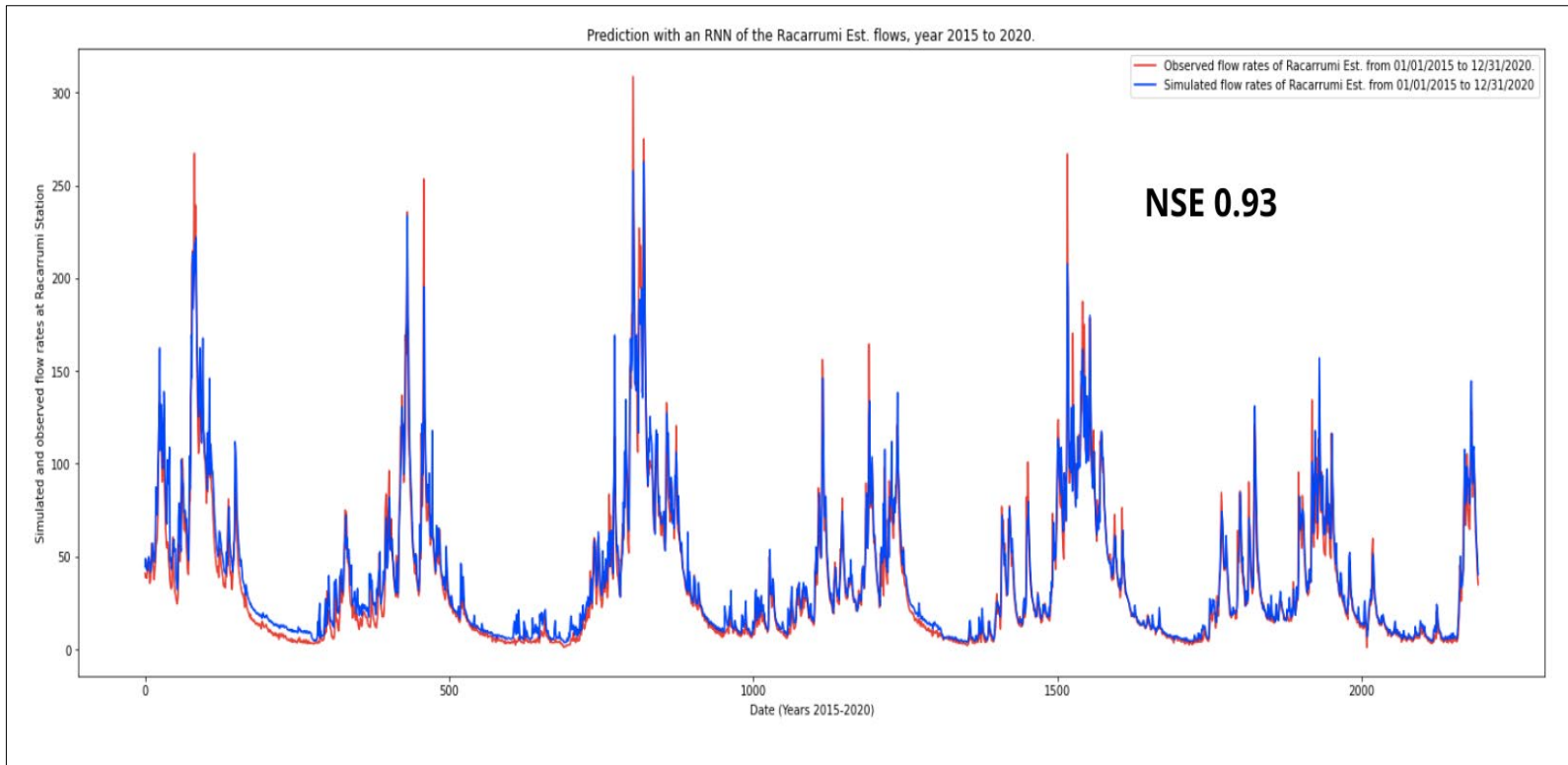


Figure 26. Hydrographs of observed flows (red) and simulated flows (blue), with an NSE index of 0.93.

In this study, the metric to evaluate the performance of hydrological modeling (Asurza, Ramos, & Lavado W, 2018) is the Nash-Sutcliffe efficiency (NSE). The procedure for calculating this metric is as follows:

$$NSE = 1 - \frac{\sum_{i=1}^n (y_i - y'_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Where:



y_i and y'_i = denote the observed and simulated runoff at time i

y = denote the average observed and simulated runoff at time i

After modeling rainfall runoff in the Chancay Lambayeque river basin, we proceeded to evaluate its performance, obtaining an NSE of 0.93, guaranteeing the reliability of the modeling, as shown Figure 27.

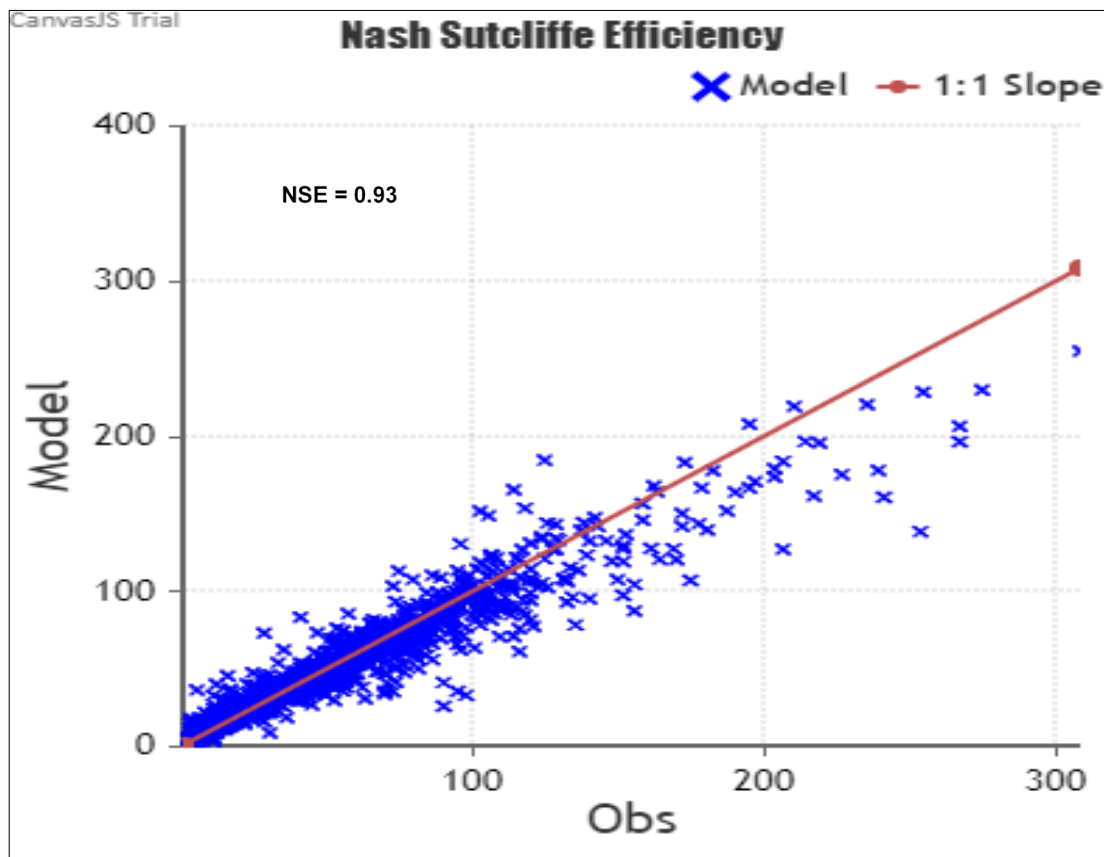


Figure 27. Scatter plot of the Nash Sutcliffe-NSE efficiency index.

Determination of flow rates for different return periods in the hydrological station "Bocatoma Racarrumi" located in the middle part of the Chancay Lambayeque river basin

We proceeded first to determine the goodness of fit using the Kolmogorov-Smirnok (SK) test, obtaining that the flow rates conform to the statistical distributions ($\Delta_{\text{theoretical}} < \Delta_{\text{tabular}}$): Normal Distribution (N), Log. Normal 2 Parameters (LN 2P), Log. Normal 3 Parameters (LN 3P), Gumbel (G) and Gamma 2 Parameters (G 2P), as shown in Table 3.

Table 3. Kolmogorov-Smirnok (SK) goodness-of-fit test for observed annual peak flows.

Models	$\Delta_{\text{theoretical}}$	Δ_{tabular}	Condition
DN	0.1087	0.2483	$\Delta_{\text{teórico}} < \Delta_{\text{tabular}}$
LN 2P	0.0943	0.2483	$\Delta_{\text{teórico}} < \Delta_{\text{tabular}}$
LN 3P	0.0958	0.2483	$\Delta_{\text{teórico}} < \Delta_{\text{tabular}}$
Gumbel	0.0762	0.2483	$\Delta_{\text{teórico}} < \Delta_{\text{tabular}}$
Gamma 2P	0.0999	0.2483	$\Delta_{\text{teórico}} < \Delta_{\text{tabular}}$

Once the goodness-of-fit analysis was performed, the design flows for different return periods were determined, for which the Gumbel methodology was chosen since it presents the lowest $\Delta_{\text{theoretical}}$ Δ with respect to the other statistical differences (Normal, LN2P, LN3P and Gumbel). In this sense, as can be seen in Table 4, there are different

flows for different return periods. It should be noted that only those obtained by the Gumbel method are presented since, if we look at Table 3, it is the one that best adjusts to the real flows, obtaining a smaller statistical difference with respect to the other models.

Table 4. Maximum flows for different return periods, calculated by the Gumbel method and plotted in Figure 28.

Nº	Tr (Years)	Q _{máx} (m ³ /s)
01	5	289.78
02	10	333.64
03	25	391.62
04	50	435.48
05	100	479.34
06	200	523.20
07	1 000	625.03

Discussion

The implementation of artificial intelligence techniques in hydrology is of utmost importance, particularly for rainfall-runoff modeling; in this sense, we share what is referred by Mosavi *et al.* (2018), who indicate that these techniques allow optimizing the results of the simulations, specifically by achieving that the simulated variables resemble the observed ones, this is also corroborated by goodness-of-fit metrics such as the Nash Sutcliffe

Coefficient shown in Figure 28, which in the present research is 0.93, i.e. the modeling is "very good".

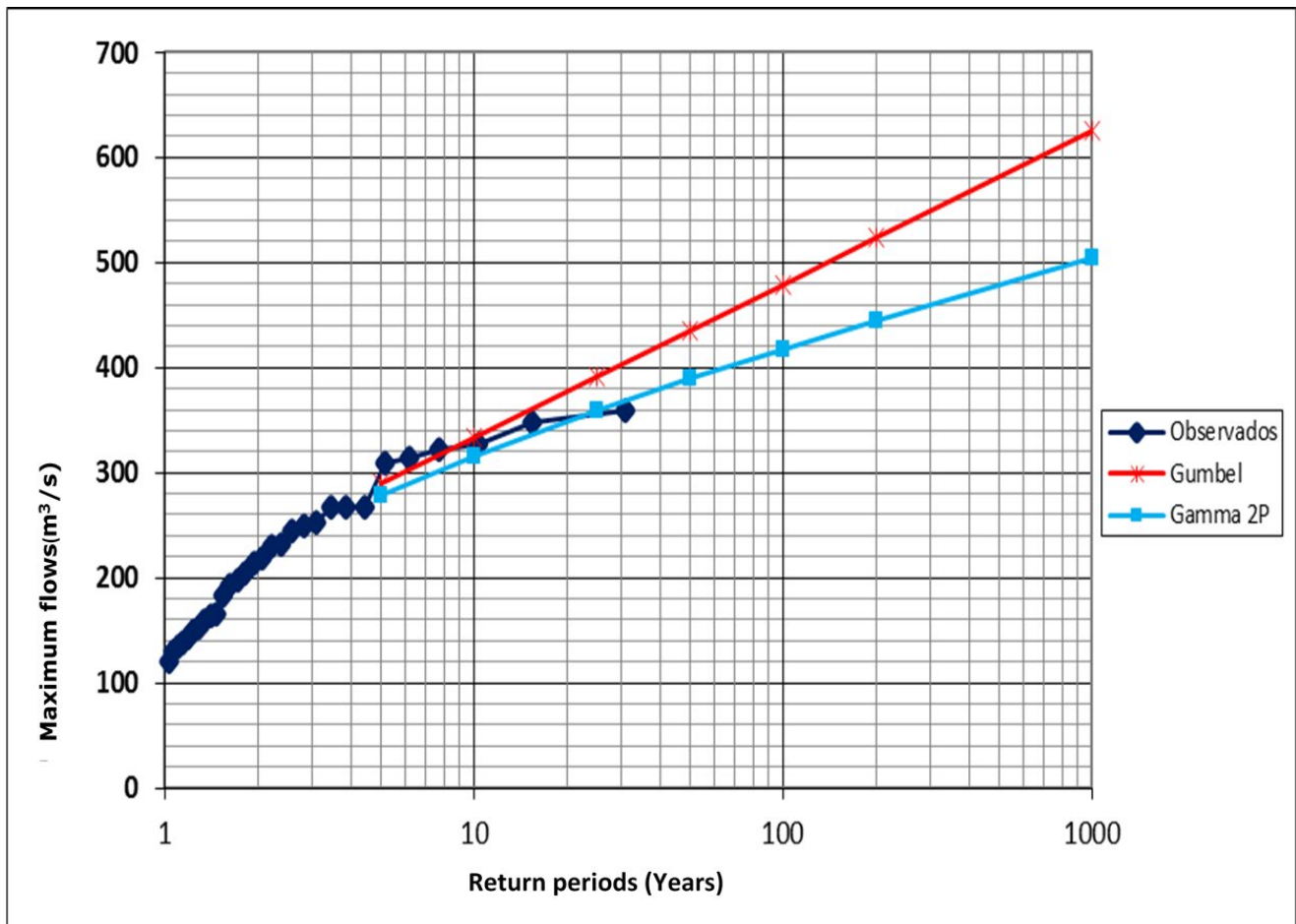


Figure 28. Design flows obtained by the statistical methods of "Gumbel and 2-parameter Gamma" for different return periods.

It should be noted that the most widely used artificial intelligence techniques for rainfall-runoff modeling and simulation of hydrological processes are artificial neural networks, as referred to by Lujano, Lujano,

Pitágoras and Lujano (2014), who indicate that with innovative programming languages they can be coded with much greater versatility, obtaining results with different types of networks and/or models that have been previously defined, namely, with ANN models that one has programmed or downloaded from servers such as Scikit-Learn or GitHub.

On the other hand, Farfán *et al.* (2020), as part of a modeling carried out with WEAP and GR2M models in the Machángara and Chulco rivers, obtained NSE coefficients of 0.64 and 0.88 respectively; these results were later processed with artificial neural networks, obtaining NSE coefficients of 0.99 in both rivers. In this line, we subscribe to what was presented by the authors, indicating that in the present research, after having carried out the modeling of rainfall runoff in the Chancay Lambayeque river basin, a coefficient close to 1.00 (0.93) was obtained.

The type of neural networks used in this research were the LSTM (Long Short-Term Memory) networks, due to their versatility and better accuracy when simulating flows, as described by Hu *et al.* (2018) in their study, who used Long and Short-Term Memory Networks (LSTM); to simulate rain-runoff in the Fen River basin (China), having obtained an NSE of 0.90 in the validation stage. For rainfall-runoff modeling in the Chancay Lambayeque river basin, records from 12 hydrometeorological stations with a total of 30 years (hydrological normal) were used; while Hu *et al.* (2018) used records from 14 stations for a period of 43 years.

Fan *et al.* (2020) indicate that among the most widely used deep learning techniques for the simulation of hydrometeorological variables, Long and Short Term Memory Networks (LSTM) stand out for their

versatility, and are currently being implemented with other techniques to generate hybrid models and thus obtain better results.

Regarding the flows obtained for different return periods, Tineo-Pongo (2018) as part of the application of a distributed hydrological model called TETIS, used 26 years of records from the hydrological station "Racarrumi", thus obtained by the Gumbel method that for a Tr of five years the maximum flow was 286.09 m³/s; for a Tr of 50 years it was 442.77 m³/s; for a Tr of 100 years it was 489.93 m³/s, and for a Tr of 1 000 years it was 646.60 m³/s. Along these lines, in the present investigation, having used 30 years of daily precipitation and flow records, it was obtained that for a 5-year Tr the maximum flow was 289.78 m³/s; for a 50-year Tr it was 435.48 m³/s; for a 100-year Tr it was 479.34 m³/s, and for a 1 000-year Tr it was 625.03 m³/s.

Finally, we agree with Young, Liu and Chung (2015) who indicate that network models allow simulating and predicting hydrometeorological variables with greater accuracy, for example, runoff simulation with automatic and/or deep learning techniques allow generating information where it does not exist, from small to large basins, so that the data generated could be used for hydraulic modeling and to define flood zones associated with different return periods.

Conclusions

This study analyzed the daily scale hydrometeorological data available in the Chancay Lambayeque river basin, namely 30 years (hydrological normal), from 01/01/1991 to 12/31/2020, where 80 % (24 years) of the

data were used for model training, while the remaining 20 % (six years) were used for validation.

Initially, the calibration and subsequent validation of the neural network model for rainfall-runoff simulation in the Chancay Lambayeque river basin was carried out using Long and Short Term Memory Networks (LSTM), thus obtaining a Nash coefficient of 0.93 in the model validation stage, corresponding to the qualification of "very good".

Table 3 shows the maximum flows determined for different return periods (Tr) at the hydrological station "Bocatoma Racarrumi" located in the middle part of the Chancay Lambayeque river basin, thus calculating flows for Tr of 05, 10, 25, 50, 50, 100, 200 and 1 000 years, as can be seen in the table.

Finally, according to the determination of flow rates for different return periods at the Racarrumi intake hydrological station, for a return period; for example, a flow rate of 435.48 m³/s for 50 years and 479.34 m³/s for 100 years. According to historical data from 1991 to 2021, the Chancay Lambayeque River has had a maximum flow of 358.54 m³/s recorded on March 27, 2009 (Record obtained at the Racarrumi station). The flows obtained from the different return periods, it is considered that the Racarrumí Intake can alleviate these flows, since this work has been designed for flows greater than 500 m³/s.

References

- Abbot, J., & Marohasy, J. (2014). Input selection and optimisation for monthly rainfall forecasting in Queensland, Australia, using artificial neural networks. *Atmospheric Research*, 138, 166-178. DOI: 10.1016/j.atmosres.2013.11.002
- Adamowski, J., Chan, H., Prasher, S., Ozga-Zielinski, B., & Sliusarieva, A. (2012). Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada. *Water Resources Research*, 48(1), 1528-1541. DOI: 10.1029/2010WR009945
- Afzaal, H., Farooque, A., Abbas, F., Acharya, B., & Esau, T. (2020). Computation of evapotranspiration with artificial intelligence for precision water resource management. *Applied Sciences*, 10(5). DOI: 10.3390/app10051621
- Asurza, F. A., Ramos, C. L., & Lavado W, S. (2018). Evaluación de los productos tropical rainfall measuring missin (TRMM) y global precipitation measurement (GPM) en el modelamiento hidrológico de la cuenca del río Huancané, Perú. *Scientia Agropecuaria*, 9(1), 53-62. DOI: 10.17268/sci.agropecu.2018.01.06
- Basagaoglu, H., Chakraborty, D., & Winterle, J. (2021). Reliable evapotranspiration predictions with a probabilistic machine learning framework. *Water*, 13(4). DOI: 10.3390/w13040557

- Béjar, W., Valeriano, K., Ilachoque, J., & Sulla, J. (2016). Predicción de caudales medios diarios en la cuenca del Amazonas aplicando redes neuronales artificiales y el modelo neurodifuso ANFIS. *Research in Computing Science*, 113(1), 23-35. DOI: 10.13053/rcs-113-1-2
- Choubin, B., Khalighi-Sigaroodi, S., Malekian, A., & Kişi, Ö. (2014). Multiple linear regression, multi-layer perceptron network and adaptive neuro-fuzzy inference system for the prediction of precipitation based on large-scale climate signals. *Hydrological Sciences Journal*, 61(6), 1001-1009. DOI: 10.1080/02626667.2014.966721
- Cromwell, E., Shuai, P., Jiang, P., Coon, E. T., Painter, S. L., Moulton, J. D., Lin, N., & Chen, X. (2021). Estimating watershed subsurface permeability from stream discharge data using deep neural networks. *Frontiers in Earth Science*, 9. DOI: 10.3389/feart.2021.613011
- Fan, H., Jiang, M., Xu, L., Zhu, H., Cheng, J., & Jiang, J. (2020). Comparison of long short term memory networks and the hydrological model in runoff simulation. *Water*, 175-190. DOI: 10.3390/w12010175
- Farfán, J., Palacios, K., Ulloa, J., & Avilés, A. (2020). A hybrid neural network-based technique to improve the flow forecasting of physical and data-driven models: Methodology and case studies in Andean watersheds. *Journal of Hydrology: Regional Studies*, 27. DOI: 10.1016/j.ejrh.2019.100652

- French, M., Krajewski, W., & Cuykendall, R. (1992). Rainfall forecasting in space and time using a neural network. *Journal of Hydrology*, 137(1), 1-31. DOI: 10.1016/0022-1694(92)90046-X
- Fu, M., Fan, T., Ding, Z., Salih, S. Q., Al-Ansari, N., & Yaseen, Z. M. (2020). Deep learning data-intelligence model based on adjusted forecasting window scale: Application in daily streamflow simulation. *IEEE Access*, 8, 32632-32651. DOI: 10.1109/ACCESS.2020.2974406
- Han, H., Choi, C., Jung, J., & Kim, H. S. (2021). Deep learning with long short term memory based sequence-to-sequence model for rainfall-runoff simulation. *Water*, 13(437).
- 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(11), 1543-1558. DOI: 10.3390/w10111543
- Jimeno, P., Senent, J., Pérez, J., Pulido, D., & Cecilia, J. (2017). Estimation of instantaneous peak flow using machine-learning models and empirical formula in peninsular Spain. *Water*, 9(5), 347-359. DOI: 10.3390/w9050347
- Kim, T., Yang, T., Gao, S., Zhang, L., Ding, Z., Wen, X., Gourley, J. J., & Hong, Y. (2021). Can artificial intelligence and data-driven machine learning models match or even replace process-driven hydrologic models for streamflow simulation?: A case study of four watersheds with different hydro-climatic regions across the CONUS. *Journal of Hydrology*, 598. DOI: 10.1016/j.jhydrol.2021.126423

- Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A., Hochreiter, S., & Nearing, G. (2019). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12), 11344-11354. DOI: 10.1029/2019WR026065
- Laqui-Vilca, W. F. (2010). Aplicación de redes neuronales artificiales a la modelización y previsión de caudales medios mensuales del río Huancané. *Revista Peruana Geo-Atmosférica RPGA*, 2, 30-44. Recovered from https://web2.senamhi.gob.pe/rpga/pdf/2010_vol02/art3.pdf
- Lujano, E., Lujano, A., Pitágoras, J., & Lujano, R. (2014). Pronóstico de caudales medios mensuales del río Ilave usando modelos de redes neuronales artificiales. *Revista de Investigaciones Altoandina*, 16(1), 89-100.
- Mosavi, A., Ozturk, P., & Chau, K.-W. (2018). Flood prediction using machine learning models: Literature review. *Water*, 10(11), 1536-1576. DOI: 10.3390/w10111536
- Nabipour, N., Dehghani, M., Shamshirband, S., & Mosavi, A. (2020). Short-term hydrological drought forecasting based on different nature-inspired optimization algorithms hybridized with artificial neural networks. *IEEE Access*, 8, 15210-15222. DOI: 10.1109/ACCESS.2020.2964584
- Qin, J., Liang, J., Chen, T., Lei, X., & Kang, A. (2019). Simulating and predicting of hydrological time series based on tensorflow deep learning. *Polish Journal of Environmental Studies*, 28(2), 795-802. DOI: 10.15244/pjoes/81557

- Rezaeianzadeh, M., Tabari, H., & Yazdi, A. (2014). Flood flow forecasting using ANN, ANFIS and regression models. *Neural Computing & Applications*, 25, 25-37. DOI: 10.1007/s00521-013-1443-6
- Rodríguez, C., Díaz, H., Ballesteros, J., Rohrer, M., & Stoffel, M. (2019). The anomalous 2017 coastal El Niño event in Peru. *Climate Dynamics*, 52, 5605-5622. DOI: 10.1007/s00382-018-4466-y
- Sattari, M., Apaydin, H., Band, S., Mosavi, A., & Prasad, R. (2021). Comparative analysis of kernel-based versus ANN and deep learning methods in monthly reference evapotranspiration estimation. *Hydrology and Earth System Sciences*, 25(1), 603-618. DOI: 10.5194/hess-25-603-2021
- Shi, L., Feng, P., Wang, B., Liu, D., Cleverly, J., & Fang, Q. (2020). Projecting potential evapotranspiration change and quantifying its uncertainty under future climate scenarios: A case study in southeastern Australia. *Journal of Hydrology*, 584. DOI: 10.1016/j.jhydrol.2020.124756
- Tineo-Pongo, P. (2018). *Aplicación del modelo hidrológico distribuido TETIS para estimar la variabilidad hidrológica en la cuenca del río Chancay Lambayeque*. Recovered from https://alicia.concytec.gob.pe/vufind/Record/UCVV_2c32a071628a390e3387d909a4a4f14f
- Valderrama-Purizaca, F. J., Chávez-Barturen, D. A., Muñoz-Pérez, S. P., Tuesta-Monteza, V., & Mejía-Cabrera, H. I. (2021). Importancia de las redes neuronales artificiales en la ingeniería civil: Una revisión sistemática de la literatura. *Iteckne*, 18(1), 71-83. DOI: 10.15332/iteckne

- Wang, J., Li, H. P., Lu, H. Y., Zhang, R. Q., Cao, X. S., Tong, C. F., & Zheng, H. X. (2019). Estimation of evapotranspiration for irrigated artificial grasslands in typical steppe areas using the METRIC model. *Applied Ecology and Environmental Research*, 17(6), 13759-13776. DOI: 10.15666/aeer/1706_1375913776
- Yaseen, Z., Naghshara, S., Salih, S., Kim, S., Malik, A., & Ghorbani, M. (2020). Lake water level modeling using newly developed hybrid data intelligence model. *Theoretical and Applied Climatology*, 141, 1285-1300. DOI: 10.1007/s00704-020-03263-8
- Young, C.-C., Liu, W.-C., & Chung, C.-E. (2015). Genetic algorithm and fuzzy neural networks combined with the hydrological modeling system for forecasting watershed runoff discharge. *Neural Computing and Application*, 26, 1631-1643. DOI: 10.1007/s00521-015-1832-0
- Zhang, Y., Zhao, Z., & Zheng, J. (2020). CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China. *Journal of Hydrology*, 588. DOI: 10.1016/j.jhydrol.2020.125087