Impact of training dataset size and its hydrometeorological typology on LSTM performance for rainfall-runoff modeling: a case study of the Severn river

Nadia Skifa, Fadil Boodoo, Carole Delenne, Renaud Hostache, Morgan Abily

IMPACT OF TRAINING DATASET SIZE AND ITS HYDROMETEOROLOGICAL TYPOLOGY ON LSTM PERFORMANCE FOR RAINFALL-RUNOFF MODELING : A CASE STUDY OF THE SEVERN RIVER

Nadia, Skifa
skifanad2@aquacloud.net

Fadil, Boodoo[1], Carole, Delenne
fadil.boodoo@umontpellier.fr, carole.delenne@umontpellier.fr

Renaud, Hostache, Morgan, Abily
renaud.hostache@ird.fr, abilmor9@aquacloud.net

KEY WORDS

Machine Learning in Hydrology, Artificial Intelligence, Univariate Analysis, Training Dataset, Conceptual Models

ABSTRACT

Accurate discharge prediction in hydrological forecasting relies on robust modeling. This study investigates the Long Short-Term Memory (LSTM) model's performance, focusing on training dataset size and hydrometeorological patterns. Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs) are also considered for spatial and temporal dependencies. Data is drawn from the CAMELS-GB dataset (1975-2015, Saxons Lode, UK). Results show that LSTM performance varies, with years surrounding high water events (like 2004) performing poorly in training, and struggles in validation. Training with one year yields 23.03% NSE values above 0.7, but using three consecutive years improves this to 84.42%. Typological differences also affect model performance. This study reveals LSTM sensitivity to training periods, aiding the optimization of training duration for better discharge prediction accuracy. Future research will delve into year selection within typological clusters.

1. INTRODUCTION

The evolution of Artificial Intelligence (AI) has significantly reshaped the field of rainfall-runoff modelling, revolutionizing conventional approaches and bolstering predictive capabilities. Over time, AI techniques, including machine learning and deep learning, have gained widespread application in rainfall-runoff modelling, facilitating the development of highly accurate and efficient models capable of learning intricate patterns from data. These AI models hold the potential to yield superior runoff predictions based on rainfall inputs, further augmented by advancements in computational power and the availability of extensive hydrological datasets. As a result, the integration of AI into rainfall-runoff modelling offers substantial promise in supporting water resource management, flood forecasting, and data-driven decision-making, ultimately contributing to more sustainable and effective water management strategies.

Rainfall-runoff modelling encompasses three broad categories: physical models, conceptual models, and AI models. Physical models adhere to the laws of physics to simulate hydrological processes in a basin, necessitating detailed basin-specific information. In contrast, conceptual models provide

[1] Corresponding author

simplified representations, emphasizing key processes and employing empirical relationships, offering computational efficiency but potentially demanding more calibration efforts. AI models, such as machine learning and deep learning, are data-driven and learn directly from historical data without explicitly incorporating physical processes. While these AI models deliver strong predictive accuracy and excel at handling complex relationships, they may sacrifice interpretability and generalization to new conditions.

Artificial Neural Networks (ANNs) have emerged as pivotal tools within the water sector, revolutionizing hydrological modelling and analysis [4,5,6]. ANNs are adept at capturing complex nonlinear relationships inherent in hydrological processes, resulting in improved predictions for rainfall-runoff modelling, flood forecasting, and water quality prediction. Their adaptability and ability to learn from historical data make them well-suited for real-time applications. ANNs have also played a crucial role in optimizing water resource management, particularly in arid regions through groundwater level prediction, underscoring their significance in advancing hydrological research and management practices.

Recurrent Neural Networks (RNNs) [7], a class of neural networks designed for processing sequential data with temporal dependencies, offer unique advantages. Unlike feedforward networks, RNNs can retain information from previous inputs [8], making them ideal for tasks involving temporal relationships. RNNs leverage a hidden state to capture long-term dependencies, enabling them to model complex temporal patterns.

Obtaining historical data can be challenging, prompting the consideration of training models with years sharing similar hydrometeorological conditions. This approach is efficient when relationships can be established. While some studies [1,2,3] have investigated the influence of the training dataset length on the LSTM model performance, this study specifically investigates the interannual variability in hydrometeorological conditions during training years, highlighting its significant influence on model performance alongside training dataset size. The study's primary objectives are to assess the impact of training dataset size on LSTM model performance and explore how the hydrometeorological conditions during training years affect the performance of the rainfall-runoff LSTM model.

2. METHODS & DATABASE

2.1 Long Short-Term Memory

Neural Network are known to be impersistent since they only receive input from the preceding iteration; as a result, Recurrent Neural Networks (RNN) came to address this issue. They can connect previous information to the present task. However, conventional RNNs are unable to learn long-term dependencies and struggle with vanishing gradients over extended sequences due to the nature of their architecture and the way gradients are propagated during training. As a consequence, Hochreiter et al. [9] introduced a special structure of RNN, the Long-Short Term Memory (LSTM) which is designed to avoid the long-term dependency problem in sequential data (Figure 1).

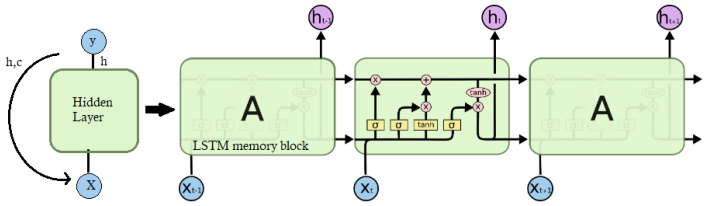

Figure 1 : Visualization of the standard LSTM model cells. The LSTM's memory block is referred to as **A** (Adapted from [10]).

In order to conduct a comprehensive analysis of the results, it is crucial to have a good understanding of the model components and functioning. Figure 2 illustrates the standard architecture of a LSTM

memory block. Here, the specific feature that allows the information to persist in RNNs, is the presence of a loop also referred to as a rolled model. In fact, the rolled model can be assimilated to a chain of identical Feed Forward ANNs, or unrolled model, as in Figure 3. These identical ANNs are characterized by having the same structure, weights and activation functions.

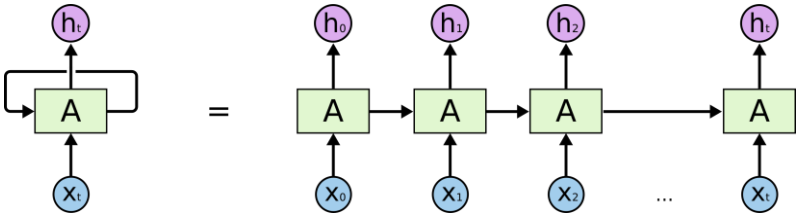

Figure 2 : On the left, a rolled Recurrent Neural Network. On the right, an unrolled Recurrent Neural Network [10].

At each time step, denoted as t, the memory block receives three types of data. Firstly, there is an input vector X_t, originating from the previous layer. This vector represents the new information at time t. Secondly, the memory block receives a hidden state vector, denoted as h_{t-1} and a cell state vector, denoted as C_{t-1}, from the hidden layer at the previous time step $t-1$. These vectors carry sequential information from the preceding time steps.

The memory block utilizes three multiplicative units, known as gates, to regulate the flow of information within the memory blocks: the forget gate f_t, the input gate i_t, and the output gate o_t. Each gate of the block is governed by a specific equation as the information moves from left to right, to control the information propagation within the LSTM architecture [11].

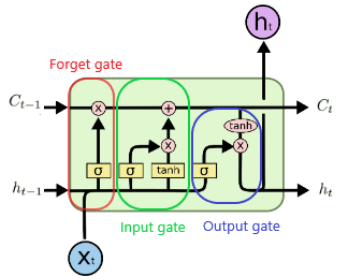

Figure 3 : LSTM memory block (Adapted from [10]).

2.1.1 The forget gate

First introduced by Gers et al. [17], the forget gate controls how much cell state information should be forgotten. It looks at h_{t-1} and x_t and attributes a number between 0 and 1 for each number in the cell state C_{t-1} (1 representing a complete keep while a 0 representing a complete delete) [10].

$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)$ (1)

Where U_f, W_f and b_f represent the adjustable matrices or vectors associated with the forget gate and $\sigma(x)$ denotes the logistic sigmoid activation function, which is defined as follows:

$\sigma(x) = \frac{1}{1+e^{-x}}$ (2)

2.1.2 The input gate

The input gate decides how much new information is taken into consideration in the cell state. It performs two steps. Initially, a sigmoid activation function determines the specific values that will be updated, calculated by equation (3) This function attributes numbers between 0 and 1 to describe how much information of each component should be let through.

$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i)$ (3)

Subsequently, a tangent hyperbolic (tanh) function generates a vector, denoted as $\widetilde{C_t}$, comprising potential new candidate values that could be incorporated into the state. In the subsequent stage, these two components are combined to produce an update for the state, as represented in equation (4)

$\widetilde{C}_t = \tanh(U_{\tilde{C}} x_t + W_{\tilde{C}} h_{t-1} + b_{\tilde{C}})$ (4)
As a result, the cell state C_{t-1} is updated to C_t using the following equation:
$C_t = f_t C_{t-1} + i_t \widetilde{C}_t$ (5)

2.1.2 The output gate

The output gate decides what information will flow into the new hidden state h_t and into the next layer through the sigmoid layer with the tanh function, as described by equations (6-7) [11].
$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o)$ (6)
$h_t = o_t \tanh(c_t)$ (7)
Knowing that we are basing the predictions on the previous n days of precipitation data, the information goes through all three gates n times before outputting predictions.

In the context of this study, the Long Short-Term Memory's architecture, is underpinned by its aptitude to effectively capture the intricate temporal dynamics and sequential dependencies intrinsic to hydrological processes, such as those encountered in rainfall-runoff modelling.
The LSTM's inherent ability to model temporal dependencies via recurrent connections is particularly relevant in hydrological modelling, aligning with the influence of past rainfall on runoff [9]. Moreover, LSTM's adaptability to variable-length sequences suits hydrological data with irregular time intervals [17]. Furthermore, its prowess in capturing intricate relationships enhances its utility for nuanced hydrological patterns [15]. While LSTM brings substantial advantages, certain considerations arise. The computational complexity of training LSTM models, especially with intricate architectures and extensive datasets, necessitates significant computational resource [18]. While LSTM mitigates vanishing gradient issues of traditional RNNs, capturing exceedingly long-term dependencies could remain challenging [19].

2.2 The conceptual hydrological model: SUPERFLEX

SUPERFLEX is a conceptual hydrological model [13], based on the robust numerical implementations of generic building pieces including reservoirs, junctions and constitutive functions. The lumped model [14] based on observed precipitation and run-off data, is composed of three reservoirs (Figure 4):

- an unsaturated soil reservoir with a storage SUR representing the upper root zone,
- a fast reservoir with storage SFR representing the fast-responding components (e.g. the riparian zone),
- a slow reservoir with storage SSR representing slow-responding components (e.g. deep groundwater).

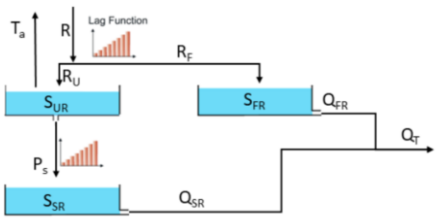

Figure 4 : SUPERFLEX Scheme [14]

Precipitation R infiltrates the soil R_U, then the UR reservoir linearly percolates into the SR reservoir (Eq. (8)), whereas surplus R_{FU} contributes directly to the S_{FR} reservoir, governed by Eq. (9). In addition, to provide for a delayed hydrological response of the basin, a triangle lag function is used ahead of the S_{FR} and S_{SR} reservoirs.
$P_S = P_{\max} \frac{S_{UR}}{S_{\max}}$ (8)
$R_F = C_r R$ and $R_U = (1 - C_r) R$ (9)
with
$\frac{1}{C_r} = 1 + \exp\left(\frac{\frac{S_{UR}}{S_c} + 0.5}{\beta}\right)$ (10)

The potential evapotranspiration T_p is converted into actual evapotranspiration according to Eq. (11):

$$T_p = T_a * \min\left(1, \frac{S_{UR}}{S_{max}} \frac{1}{L_p}\right) \quad (11)$$

where L_p is the fraction of S_{max} below which T_p is constrained by S_{UR}.

The discharges through the reservoirs S_{FR} and S_{SR} are calculated according to the time scales K_{FR} and K_{SR}, respectively.

$$Q_{SR} = \frac{S_{SR}}{K_{SR}} \quad \text{and} \quad Q_{FR} = \frac{S_{FR}}{K_{FR}} \quad (12)$$

The SUPERFLEX model is characterized by 6 state variables and composed of 8 parameters, as described in Table 1.

Table 1 : Variables and parameters of the SUPERFLEX model

	Parametres	Units	Meaning
State variables	S_{UR}	mm	UR storage reservoir
	S_{FR}	mm	FR storage reservoir
	S_{SR}	mm	SR storage reservoir
	Q_{UR}	mm/h	UR discharge
	Q_{FR}	mm/h	Fast discharge
	Q_{SR}	mm/h	Slow discharge
Parametres	t rise	1/h	Hydrograph lag time
	K_{FR}	1/h	FR time scale
	K_{SR}	1/h	SR time scale
	K_{UR}	1/h	UR outflow rate scale
	S_{max}	mm	Maximum UR storage
	Beta	-	Limit for PET
	alpha Fr	-	FR power coefficient
	alpha Sr	-	SR power coefficient

The modelling using the SUPERFLEX model presents distinctive advantages and challenges within the realm of hydrological modelling. One prominent advantage lies in its grounding in hydrological concepts, affording the explicit integration of domain expertise and hydrological processes. This characteristic enhances model interpretability, facilitating insights into the underlying mechanisms governing hydrological behavior. Fenicia et al. [15] emphasize the potency of conceptual models like SUPERFLEX in elucidating catchment response. However, this advantage is coupled with the potential demand for meticulous parameter calibration, a task that can prove labor-intensive and intricate due to the non-linear interactions inherent in hydrological systems. Furthermore, the SUPERFLEX model might encounter difficulties in capturing complex nonlinear relationships existing within observed data, potentially resulting in diminished predictive accuracy in scenarios where such relationships are pivotal. Moreover, Wagener et al. [16] expound on the challenges tied to model complexity and the associated parameter estimation. In summation, while the SUPERFLEX model offers a physically-informed framework and heightened interpretability, its efficacy is contingent upon robust parameterization and the accommodation of intricate hydrological interactions.

Through a comparative approach, the study's results involve a direct comparison between LSTM and the SUPERFLEX model. Where LSTM leverages data-driven learning, SUPERFLEX represents a conceptual model founded on domain-specific knowledge. The interpretability of SUPERFLEX enables insights into underlying hydrological processes [20]. However, its success relies on accurate parameterization and capturing complex nonlinear relationships within data [21]. In the Severn River study, the comparative evaluation of LSTM's data-driven approach against SUPERFLEX's physically-informed methodology will provide valuable insights into their respective capabilities and limitations, contributing to advancing hydrological modelling practices.

2.4 Evaluation metrics

In order to evaluate the efficiency and the robustness of both the LSTM and SUPERFLEX models, different efficiency criteria are available to quantify the adequacy between simulations and observations, such as the Nash–Sutcliffe efficiency [22].

Nash–Sutcliffe efficiency (NSE)
The NSE is the most used metric in hydrology for its ability to normalize the model's performance[29] sing the equation (13).

$$\mathrm{NSE} = 1 - \frac{\sum_{t=1}^{T}(Q_{\mathrm{sim}}^{t} - Q_{\mathrm{obs}}^{t})^{2}}{\sum_{t=1}^{T}(Q_{\mathrm{obs}}^{t} - \overline{Q_{\mathrm{obs}}})^{2}} \quad (13)$$

where T is the total number of time steps, Q_{sim}^{t} the simulated discharge at time t, Q_{obs}^{t} the observed discharge at time t, and $\overline{Q_{\mathrm{obs}}}$ the mean observed discharge. We differentiate between three values:

- NSE=1 indicates a perfect match between model simulations and observed data
- NSE=0 suggests that the model's predictive power is equivalent to the mean of the observed data
- NSE < 0 indicates that the model performs less than the mean of the observed data

A significant drawback of the Nash-Sutcliffe efficiency metric is its reliance on squared differences between observed and predicted values. This calculation method leads to an overweighting of larger values in a time series, while lower values tend to be overlooked or downplayed.

In this study, the data from Saxons Lode gauging station is used to characterize the Severn riven upstream of its confluence with the Avon river.

2.5 Study area: River Severn at Saxons Lode gauge

The Severn River, running from its origin in the Welsh highlands of Plynlimon to the Bristol Channel, holds the distinction of being the longest river in England, with a total length of approximately 354 km. While urban centers like Worcester, Tewkesbury and Evesham are scattered along its course, the majority of the catchment area, which spans approximately 11,000 km^2, is rural in nature, making it the largest river basin in England. Figure 5 displays the catchment boundaries and river network, along with the position of the accessible gauging station of Saxons Lode at the Severn River. Moreover, the Severn basin is a vital source of water for drinking, irrigation and industry; it also supports a range of wildlife, including fish, birds and mammals[25].

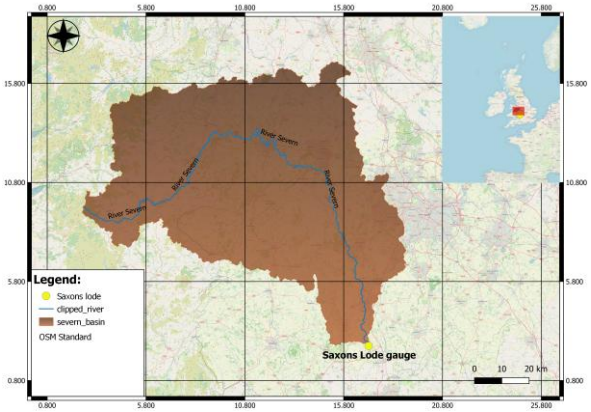

Figure 5 : Map of the Severn river: catchment boundaries and location of the Saxons Lode gauging station

The focus of the study lies in the region near the Saxons Lode gauging station, upstream of the confluence between the Severn and Avon rivers. This area, has been prone to recurrent flooding due to intense rainfall, with significant floods recorded in 1947, 2000, 2007 and 2012 among others [26].

2.6 CAMELS-GB database

In Great Britain, open-source datasets including quality-controlled river flow data are readily available through the UK National River Flow Archive (NRFA) [23], there has been a lack of integration and standardized processing of these datasets across a consistent set of catchments. This has prevented the creation of a centralized resource that can be easily accessed by the public.
Furthermore, these datasets are subject to constant changes, making it difficult to conduct consistent and repeatable analyses. Additionally, their range of variables and catchment attributes is more limited compared to larger-sample datasets like CAMELS (Catchment characteristics and MEteorology for Large-Sample Studies) [24].

To address this data gap, the CAMELS-GB dataset was developed. It combines hydrometeorological time series and catchment characteristics for 671 catchments throughout Great Britain. It is a high-quality, large-sample, watershed-scale hydrometeorological dataset that provides data for over a wide range of climatic, hydrological, landscape and human management characteristics between 1970 and 2015.

Daily time series of hydro-meteorological data including rainfall, potential evapotranspiration, humidity and river flow. Additionally, other attributes such as topography, land cover, soils and human management are quantified. This publicly available and easily accessible dataset also provides estimates of discharge uncertainty.

CAMELS started as a project to offer hydro-meteorological time series for the continuous United States as well as catchment characteristics comprising climatic indices, hydrologic signatures, land cover, soil and geology. Since then, the dataset has served as a valuable resource for researchers and practitioners in hydrology and related fields. Its large sample size and wide range of attributes make it a useful tool for analyzing and modelling water resources in Great Britain. The dataset's estimates of discharge uncertainty also provide a valuable resource for assessing the reliability and reusability of hydrological models and predictions. Overall, the CAMELS-GB dataset is a comprehensive and accessible resource for studying and managing water resources in Great Britain.

3. MODEL SET-UP AND IMPLEMENTATION

3.1 General methodology

In order to investigate the influence of the learning data size on the performance of the prediction accuracy, multiple experiments were carried out with various data sizes for training.
Two different rainfall-runoff models, an AI model (LSTM) and a conceptual model (SUPERFLEX), were compared and put to the test for predicting discharge time series. While the implementations of each method differ, they both use historical data to build models that can predict the discharge with a sufficient level of accuracy. We study the **Impact of the training data size on the performance of the LSTM Vs. SUPERFLEX.** In an attempt to study how data length affects model efficiency, the learning phase is conducted using six variants. The first 39 years of datasets are used to produce the six versions that were analysed in the training process, and the last 10 years of the dataset serve for validation. The training periods are taken successively from 1976 in bands of 1, 3, 6, 9, 12 and 15 years.

To avoid the problem of overfitting that may occur when trying to maximize the performance of the model, instead of implementing a training/test split, a training/validation/test split is used. The objective of this method, also known as the holdout method, is to compute the network's error estimation after each iteration of validation data that hasn't been seen, and to halt training when the validation data's error rate starts to rise.

The training dataset is considered to start from 1976 due to the presence of missing precipitation values in the dataset from 1971-1975.

3.2 Data pre-analysis

One key aspect before starting rainfall-runoff modelling, is to first analyze the input data (precipitation and discharge). Possible precipitation trends and metadata can help set logical interpretations of the model results when simulating discharge.
Daily precipitation and discharge data of 45 years (1971-2015) were extracted from the CAMELS-GB database for the Saxons Lode station at the Severn River (see section 2.5).

To statistically assist the monotonic trends in the precipitation time series, the Mann-Kendall test [27] is performed. It's a non-parametric test, which means that the data doesn't have to follow a normal distribution.
The test is based on two hypotheses [27]:
- The null hypothesis H_0 asserts that the time series is independently distributed.
- The alternative hypothesis H_1 states that there is a monotonic trend for the provided time series.

The Mann-Kendall test yields a p-value of 0.1923, along with a test statistic of 1.304 and a Kendall's Tau value of 0.007.
Overall, based on the obtained results in table 2, there is no significant trend observed in the precipitation data. The p-value is relatively high, indicating that the observed trend is likely due to random variation rather than a systematic pattern.

3.3 Model Set-up

To set up the models, the chosen programming language for the study is Python 3.10. In addition, common libraries were used such as Numpy for working with numerical data, Pandas for data analysis tasks, and Scikit-Learn for data postprocessing. Additionally, Matplotlib and Seaborn were used to represent the results in graphics. Finally, due to the high computation of the models, a high performance computing (HPC) system, Grid5000 was used to run the code. This large-scale and flexible testbed allows expensive computational experiments to be ran in a parallel or distributed computing including Cloud, HPC, Big Data and AI.

3.3.1 LSTM

NeuralHydrology: Experimental design

In this section, the LSTM model set up is introduced, in more details, following the structure that was adopted in the NeuralHydrology library [28].
With a strong emphasis on hydrological applications, NeuralHydrology is an open source Python library built on PyTorch that is dedicated to the development, use, and experiment with Deep Learning models. Pre-built models and data loaders enable quick experiments, but the framework is also easily extensible to new models, data sets, loss functions, or metrics to suit more sophisticated use-cases. The library was generalised and made available to anyone.
The library was created for newcomers with minimal programming expertise in mind, which made its use practical and swift. For instance, NeuralHydrology enables the training of cutting-edge rainfall-runoff models by only modifying a configuration file and without the use of any code.
In the given study, the memory consists of seven types of dynamic data, namely precipitation, potential evapotranspiration, temperature, humidity, shortwave radiation, longwave radiation and wind speed. The output gate comprises the predicted volume discharge and specific discharge.

Hyperparameters

In addition to the adjustable matrices mentioned in section 2.1, there are various hyperparameters that need to be determined to define the structure and training characteristics of the LSTM model. These hyperparameters play a crucial role in shaping the LSTM's behavior. For instance, the number of hidden layers and the number of neurons within each hidden layer determine the overall structure of the LSTM. In this particular study, the LSTM architecture consisted of two hidden layers. The model was fine tuned via a trial and error approach using different values of hidden layer size, namely 2, 4, 8, 16, 32, 64 and 128, along with different sequence lengths of 30, 90, 180, 270 and 360.

Table 2 : Configuration of the LSTM model used in this study

Hyperparameter	Meaning	Value
Layers	The layer that seperates the input and output, in which the function gives the inputs weights and sends them through an activation function as the output.	2
Activation function	An activation function determines whether a node's output is ON or OFF. By adding non-linearity to models, these functions enable the model to learn non-linear prediction bounds.	Relu
Optimiser	It adjusts the learning rate during training by reducing the learning rate according to a pre-defined schedule, here the "Adaptive learning rate" is used.	Adam
Hidden cells size	The number of neurons per hidden layer, this along with the number of layers decides whether the model is more likely to under/over-fit the data.	64
Epochs	This determines the number of full dataset iterations to be conducted; it should be increased until the validation accuracy starts to drop while the training accuracy rises.	50
Dropout	A layer that reduces the sensitivity to particular weights of the individual neurons and prevents overfitting in training by avoiding randomly chosen neurons.	0.4
Exprimental runs	A number of runs that insure that the model's variance is taken into consideration.	30
Batch size	It defines the number of samples to work on before the internal parameters of the model are updated.	300
Loss function	A function that analyses how effectively the neural network represents the training data by comparing the target and predicted output values.	MSE

Implementation

In this study, in order to evaluate the model performance for each training and validation year, the required parameters are contained in a YAML configuration file called "basin.yml", which defines the

training and validation periods. The year values in the "basin.yml" file are automatically changed for each training year in the dataset, and the model is trained 30 times to account for stochastic variance. The model's performance is assessed using the evaluation run during training, and NSE values are shown for each run to track convergence.

For each iteration of the validation process, 30 sets of simulated discharge values are produced, encompassing the whole validation period. A histogram is created utilising the NSE values obtained from each simulation in order to integrate the results of the 30 runs. The "highest point" on the histogram, which denotes the maximum or peak frequency of NSE values, is use for further analyses. This peak acts as representative measure of the total outcome.
The observed and simulated discharges are shown throughout the whole time period for each validation year to show the model's performance.

Each training and validation year's NSE values are recorded in a CSV file and then postprocessed to get a heatmap and other statistics. This NSE matrix enables a thorough analysis of modelperformance throughout several training and validation years. The final outcomes show how reliable the model is and how well it can reproduce the temporal dynamics of the discharge time series.

3.3.2 SUPERFLEX

The SUPERFLEX model structure is built in the FLEX framework [14] utilising generic parts meant to simulate its numerical functions, as enumurated below:
1. Reservoir element: represents the storage and release of water.
2. Lag function element: reflects the transmission and delay of fluxes.
3. Junction element: depicts the splitting, merging and/or rescaling of fluxes.

The main pillars of several extant conceptual models are these elements. These components may be organised into various flow configurations in SUPERFLEX, where they are generalised to reflect various conceptual assumptions of catchment function.
The hydrological model's inputs were obtained from the CAMELS-GB dataset as for the LSTM model (e.g., rainfall and 2 m air temperature and potential evapotranspiration data). The model was calibrated using data spanning from 1976 to 2004. The calibrated model's performance was then validated for the period from 2005 to 2014.
All 8 parameters of the SUPERFLEX model are calibrated using a Monte Carlo approach. For each parameter, it defines behavioural intervals and iteratively generates parameter sets within these intervals. To achieve this, each parameter's random value is generated by our Python code at predetermined intervals and used in the SUPERFLEX model to simulate the discharge time series. The corresponding observation at the gauging stations on the same day is compared with simulated value and the NSE is therefore computed. In order to choose the ideal parameter set, 10.000 simulations are performed. The parameter set yielding the best NSE values is selected as optimal.

4. RESULTS & DISCUSSION

In this section, the outcomes of our experiments are presented, focusing on the effects of varying training data size and the influence of hydrometeorological typology on the performance of the LSTM and SUPERFLEX models.
To elucidate this, the size of training data was systematically adjusted while maintaining constant values for other parameters. The models was trained over various training duration, specifically 1, 3, 6, 9, 12 and 15 years, within the training period from 1976 to 2004. Then, the model’s performance was evaluated for each individual year within the validation period, between 2005 and 2014.
The outcomes reveal a notable trend in how these models react to alterations in the training dataset's magnitude.

The variation of runoff (observed and simulated) in the Severn River for the year 2010 is provided in Figure 8 in order to illustrate the impact of the learning data size on the hydrograph components of both the LSTM and the SUPERFLEX models. The year 2010, represents the validation year with the lowest performance of the SUPERFLEX model. The training periods represented in the figure are the first periods of each training case (for instance for 3 years the period is 1976-1978, and for 6 years the period is 1976-1981...).

The learning data size is generally increased to provide more stability. However, when increasing the data size to 15 years, both models fail to capture almost all the peaks.
When comparing the two models, it becomes evident that the LSTM model outperforms SUPERFLEX in recreating Q_{obs} (observed discharge) when increasing the training data size. Specifically, for the 12 and 15 years training data durations, the LSTM model consistently exhibits superior predictive capabilities in replicating Q_{obs} values. This behavior can be attributed to the model's inherent capacity to exploit larger datasets for learning intricate dependencies within the temporal data. These observations are consistent with the existing literature, it is conventionally anticipated that the performance of the LSTM model will either reach a plateau after a certain threshold of training data or exhibit an enhancement as the training dataset size is increased.

In contrast, SUPERFLEX demonstrates a more consistent behavior across the various training data sizes, as evidenced by its similar performance for the year 2010 across the range of training data sizes examined.
This highlights a key distinction in the models' responses to training data sizes: while LSTM benefits from an increased training data history, SUPERFLEX maintains a relatively stable performance regardless of the data volume.

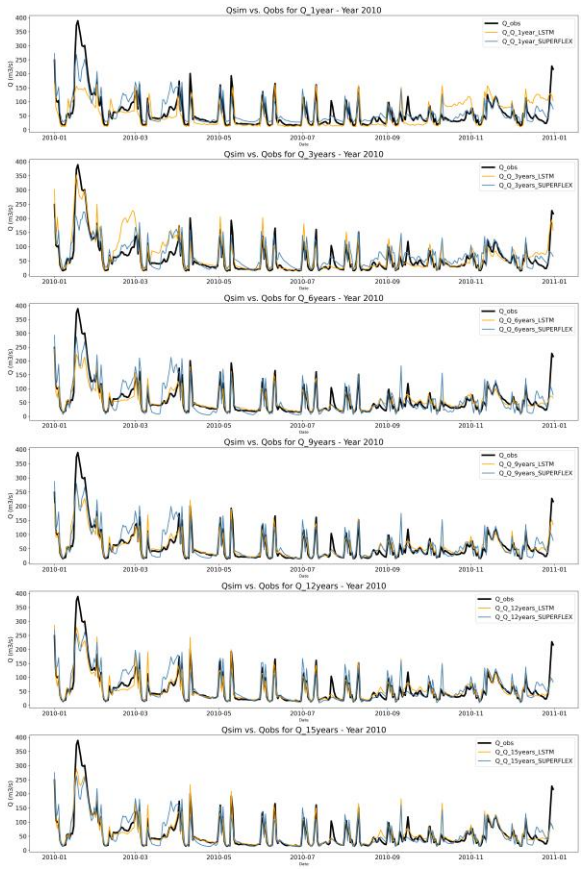

Figure 6 : Hydrograph of observed and predicted discharge of the year 2010 for different training data size

In Figure 11 and 12, The x-axis represents the training years and the y-axis contains the validation years. The color grading indicates the NSE range: the greener the matrix cells, the higher the NSE value.

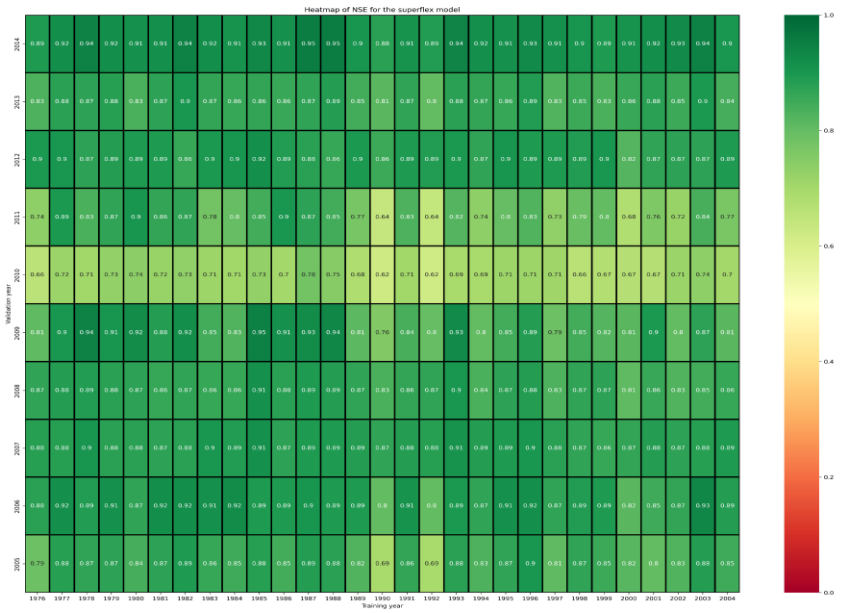

Figure 7 : 1-year training NSE heatmap for the SUPERFLEX model

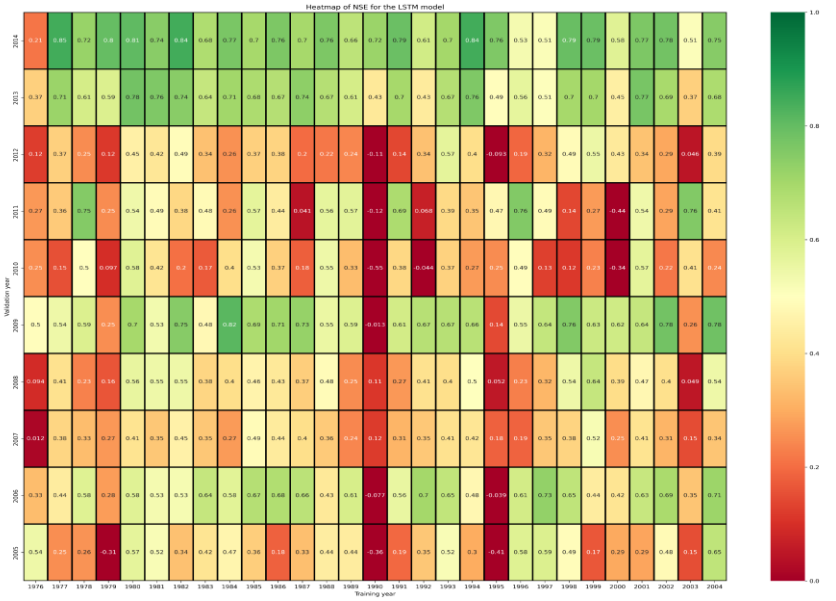

Figure 8 : 1-year training NSE heatmap for the LSTM model

The LSTM model suggests significant fluctuations compare to SUPERFLEX model for 1 training year, this could be explained by its sensitivity to data availability.

In Figure 9, a visual representation of the results of the learning processes for the LSTM and SUPERFLEX approaches is shown (the y-axis values are different for each subplot for better representation of the boxes). The data duration used to train both models ranged from 1 to 15 years. In terms of NSE statistics, there is a significant difference between the two models under investigation. Analyzing the box plots, indicates that there are two aspects that need to be addressed: the whiskers of the boxes and the presence of outliers. On the one hand, when training the LSTM model for 3 and 6 years, there is a high variability in the NSE values, which may indicate an inconsistent performance of the model. On the other hand, increasing the training size to 12 and 15 years, shortens the whiskers and improves the values of the median NSE. Conversely, the SUPERFLEX model reveals many outliers,

in all test cases, this suggests that its performance varies across the different validation years. Accordingly, the model reveals a unique behaviour when validated against the year 2010, here represented by the outliers. This can be explained by the low number of simulations (10000 simulations) compared to the number of parameters to calibrate (8 parameters).
The 1, 3, and 6 years of training data are the tests where the SUPERFLEX model outperforms the LSTM model. Numerous presumptions might account for such scenarios, with small amount of data being the main reason for this modelling case. However, the most rationale reason behind SUPERFLEX's superiority over LSTM in scenarios with limited data (such as 1 year) is attributed to the significantly higher parameter count within LSTM compared to SUPERFLEX. Notably, SUPERFLEX possesses a substantially reduced parameter set, allowing for effective parameter determination even with modest data availability. Once established, this parameter configuration reaches a state of relative optimality, leading to a stagnation in performance enhancement as observed in SUPERFLEX. This characteristic elucidates why incremental increases in the training period do not notably enhance SUPERFLEX's performance. In contrast, the LSTM model, with its larger parameter space, necessitates a more substantial volume of data to identify its optimal configuration. In essence, this phenomenon is predominantly an issue of aligning the model's degrees of freedom (parameters) with the constraints imposed by the training data, rather than merely a matter of encountering local minima.
In the 15 years training data size the NSE values stagnate in both models with a slight difference compared to the 12 years training data size matrix. This could imply that the models reach a stage of saturation, where additional training data doesn't significantly improve their predictive performance. Further investigation needs to be conducted to confirm weither the models will reach a plateau of NSE values, and that an increase in the training years will have low to no impact on the performance.

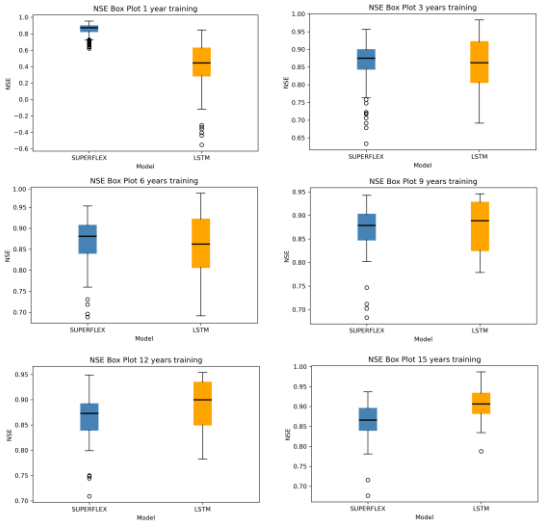

Figure 9 : Box plot of NSE for LSTM and SUPERFLEX prediction for the each test period.

For additional insights through the performances of both models, Figure 10 provides more indepth comparison between the median, min-max and number of outliers of the LSTM and SUPERFLEX models. The LSTM exceeds the SUPERFLEX model and is resilient to all data learning sizes. As training dataset length increases from 1 to 6 years, the median NSE for the LSTM model quickly rises. Later, the growth rate slows as the median NSE gets closer to a value of one. However, SUPERFLEX's median slightly decreases once it surpasses 6 years of training data.
The first LSTM's NSE statistics (medians, min-max and number of outliers) are slightly higher in the span of 9 to 15 years than those of the second model. For instance, the LSTM's medians vary from 0.86 to 0.92, whereas the SUPERFLEX's medians are in the range [0.86-0.88].
Plotting the difference between the maximum and minimum NSE values for each training year in relation to the total number of training years is shown in Figure 10. For a particular number of training years, this difference measures the consistency of NSE across distinct validation years and training years. Throughout the validation years, we can observe a gradual improvement in the consistency of both models' performance. The difference in NSE between the poorest and best validation years is about 0.2 after 15 years of training. A median NSE of 0.91 is attained by the LSTM model after 15 years of training, surpassing SUPERFLEX by the time the number of training years reaches 6 years. Figure 10

shows the number of outliers in respect to the training years. Although the number of outliers for the SUPERFLEX model is larger after 1 year of training, this count rapidly drops after 6 years and finally drops to zero after 15 years, suggesting better consistency in the model's performance over time. This observation reveals that the decreasing frequency of outliers, which indicates improved consistency in the model's performance over time, is an indication of how the SUPERFLEX model benefits from extensive training.

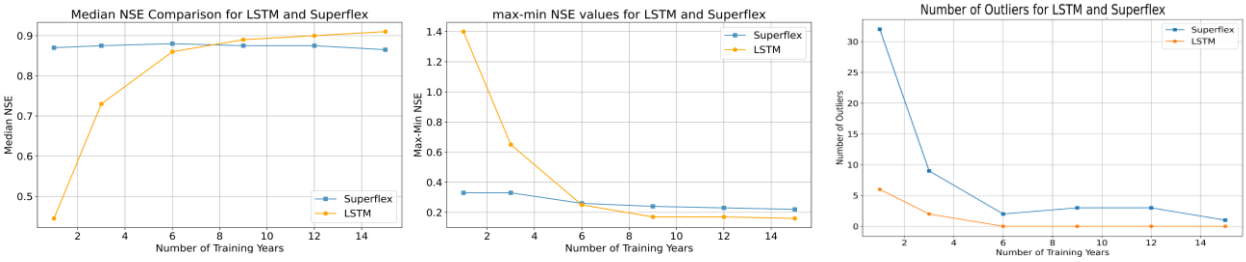

Figure 10 : NSE metrics in respect to the training data size. On the left figure the median of NSEs, in the middle figure max-min NSE values and on the right figure number of outliers of NSEs, with respect to the training data size.

5. CONCLUSION AND FUTURE WORK

In conclusion, this research provides significant insights into the performance dynamics of the LSTM model in the context of rainfall-runoff modelling. The study demonstrates a clear relationship between training data size and model performance, highlighting the positive impact of larger training datasets on LSTM's predictive capability. Notably, the performance improvement saturates beyond a certain training data size, suggesting the existence of a plateau beyond which additional data may yield marginal benefits, which calls for further investigation.

Building upon the findings of this research, several avenues for future investigations emerge. Future research could delve deeper into the intricate relationships between model performance, training data characteristics, and hydrometeorological typology, ultimately refining the predictive capabilities in the domain of hydrological modelling. Further analysis could offer a more explicit relationship between the hydromeorological typology of the years and the LSTM model. Additional tests should be conducted, such as selecting a different validation period, other training years combinations of the same cluster, and choosing a different dataset other than CAMELS.

Additionally, extending the study to encompass a broader range of climatic and hydrological contexts might enhance the generalizability of the conclusions. Incorporating advanced techniques, such as attention mechanisms or ensemble methods, could refine predictive accuracy even further. Moreover, investigating the scalability and transferability of the identified trends to different geographical regions or watersheds could provide a broader perspective on the models' adaptability. Ultimately, this research lays the groundwork for a dynamic array of future endeavors aimed at refining hydrological modelling techniques and unraveling the intricate interplay between model architecture, training data, and environmental factors.

REFERENCES AND CITATIONS

[1] T. Boulmaiz, M. Guermoui, and H. Boutaghane. (2020). *Impact of training data size on the lstm performances for rainfall–runoff modeling*. Modeling Earth Systems and Environment, 6(4):2153 2164

[2] Anbang Peng, Yuanyang Tian, Wei Xu, and Xiaoli Zhang. (2022). Effects of Training Data on the Learning Performance of LSTM Network for Runoff Simulation.

[3] Jonas Gütter, Anna Kruspe, Xiao Xiang Zhu, and Julia Niebling. (2022). *Impact of training set size on the ability of deep neural networks to deal with omission noise*. Frontiers in Remote Sensing.

[4] Holger R. Maier and Graeme C. Dandy. (1996). *The use of artificial neural networks for the prediction of water quality parameters.* Water Resources Research, 32:1013–1022

[5] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. (2014). *Neural Network Design*. PWS Publishing Company

[6] Noor-E-Ashmaul Husna, Sheikh Hefzul Bari, Md. Manjurul Shourov, M Tauhid Ur Rahman, and Mashrekur Rahman. (2016). *Ground water level prediction using artificial neural network*. International Journal of Hydrology Science and Technology, 6:371– 381.

[7] Shun-ichi Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. (1972). IEEE Transactions on Computers, C-21:1197–1206

[8] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. (2018). *State-of-the-art in artificial neural network applications: A survey*. Heliyon, 4(11):e00938

[9] Sepp Hochreiter and Jürgen Schmidhuber. (1997). *Long Short-Term Memory*. Neural Computation, 9(8):1735–1780.

[10] Christopher Olah. Understanding lstm networks. (2017). *https://colah.github.io/posts/2015-08-Understanding-LSTMs*. Accessed: 2023-05-23.

[11] Liao Weihong, Lei Xiaohui, Wang Ruojia, and Lei Xiaohui. *Rainfall-runoff modelling based on long short-term memory (lstm)*. (2019). In Proceedings of the 38th IAHR World Congress, pages 5411–5420, Panama. IAHR.

[12] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. (2000). *Learning to Forget: Continual Prediction with LSTM*. Neural Computation, 12(10):2451–2471.

[13] Fabrizio Fenicia, Dmitri Kavetski, and Hubert H. G. Savenije. (2011). Elements of a flexible approach for conceptual hydrological modeling: 1. motivation and theoretical development. Water Resources Research, 47(11).

[14] Concetta Di Mauro, Renaud Hostache, Patrick Matgen, Ramona Pelich, Marco Chini, Peter Jan van Leeuwen, Nancy Nichols, and Günter Blöschl. (2022). *A tempered particle filter to enhance the assimilation of sar-derived flood extent maps into flood forecasting models.* Water Resources Research, 58(8):e2022WR031940. e2022WR031940 2022WR031940.

[15] Fabrizio Fenicia, Dmitri Kavetski, and Hubert H. G. Savenije. (2011). Elements of a flexible approach for conceptual hydrological modeling: 1. motivation and theoretical development. Water Resources Research, 47(11).

[16] T. Wagener, D. P. Boyle, M. J. Lees, H. S. Wheater, H. V. Gupta, and S. Sorooshian. (2001). A framework for development and application of hydrological models. Hydrology and Earth System Sciences, 5(1):13–26.

[17] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. (2014). *On the properties of neural machine translation: Encoder-decoder approaches*. arXiv preprint arXiv:1409.1259.

[18] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. *An empirical exploration of recurrent network architectures*. (2015). In Francis Bach and David Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 2342–2350, Lille, France. PMLR.

[19] Y. Bengio, P. Simard, and P. Frasconi. (1994). *Learning long-term dependencies with gradient descent is difficult*. IEEE Transactions on Neural Networks, 5(2):157–166

[20] K. Beven*. How far can we go in distributed hydrological modelling? (2001). Hydrology and Earth System Sciences, 5(1):1–12.

[21] Martyn P. Clark, Dmitri Kavetski, and Fabrizio Fenicia. (2011). *Pursuing the method of multiple working hypotheses for hydrological modeling*. Water Resources Research, 47(9).

[22] J.E. Nash and J.V. Sutcliffe. (1970). River flow forecasting through conceptual models part i — a discussion of principles. Journal of Hydrology, 10(3):282–290.

[23] NRFA API. https://nrfaapps.ceh.ac.uk/nrfa/nrfa-api.html. Accessed: 2023-05-22.

[24] A. J. Newman, M. P. Clark, K. Sampson, A. Wood, L. E. Hay, A. Bock, R. J. Viger, D. Blodgett, L. Brekke, J. R. Arnold, T. Hopson, and Q. Duan. (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*. Hydrology and Earth System Sciences, 19(1):209–223.

[25] Natural Resources Wales Environment Agency. (2025). Part 1: Severn river basin district river basin management plan.

[26] Vita Ayoub, Carole Delenne, Marco Chini, Pascal Finaud-Guyot, David Mason, Patrick Matgen, Ramona Maria-Pelich, and Renaud Hostache. (2022). *A porosity-based flood inundation modelling approach for enabling faster large scale simulations*. Advances in Water Resources, 162:104141.

[27] Henry B Mann. (1945). *Nonparametric tests against trend*. Econometrica: Journal of the econometric society, pages 245–259.

[28] Frederik Kratzert, Martin Gauch, Grey Nearing, and Daniel Klotz. (2022). *Neuralhydrology — a python library for deep learning research in hydrology*. Journal of Open Source Software, 7(71):4050.

[29] P. Krause, D. P. Boyle, and F. Bäse. (2005). Comparison of different efficiency criteria for hydrological model assessment. Advances in Geosciences, 5:89–97.