Preprints.org

Article

Deep Learning-Based River Flow Forecasting with MLPs: Comparative Exploratory Analysis Applied to the Tejo and the Mondego Rivers

Gonçalo Jesus [*] , Zahra Mardani , Elsa Alves , Anabela Oliveira

Article

Deep Learning-Based River Flow Forecasting with MLPs: Comparative Exploratory Analysis Applied to the Tejo and the Mondego Rivers

Gonçalo Jesus *,† [ID], **Zahra Mardani** † [ID], **Elsa Alves** [ID], and **Anabela Oliveira** [ID]

Laboratório Nacional de Engenharia Civil, Lisboa, Portugal

* Correspondence: gjesus@lnec.pt
† These authors contributed equally to this work.

Abstract: This paper presents an innovative service for river flow forecasting and its demonstration in two dam-controlled rivers in Portugal, Tejo, and Mondego rivers, based on using Multilayer Perceptron (MLP) models to predict and forecast river flow. The main goal is to create and improve AI models that operate as remote services, providing precise and timely river flow predictions for the next 3 days. This paper examines the use of MLP architectures to predict river discharge using comprehensive hydrological data from Portugal's National Water Resources Information System (Sistema Nacional de Informação de Recursos Hídricos, SNIRH), demonstrated for the Tejo and Mondego river basins. The methodology is described in detail, including data preparation, model training, and forecasting processes, and provides a comparative study of the MLP model's performance in both case studies. The analysis shows that MLP models attain acceptable accuracy in short-term river flow forecasts for the selected scenarios and datasets, adeptly reflecting discharge patterns and peak occurrences. These models seek to enhance water resource management and decision-making by amalgamating modern data-driven methodologies with established hydrological and meteorological data sources, facilitating better flood mitigation and sustainable water resource planning as well as accurate boundary conditions for downstream forecast systems.

Keywords: river flow forecasting; artificial intelligence; deep learning; MLP; SNIRH

1. Introduction

Predicting river flow is essential for effective and sustainable management of water resources, including timely flood emergency warnings [1]. Accurate and reliable river flow forecasts are crucial for a variety of sectors such as agriculture, water resource management, flood emergency services, and hydroelectric power production [2]. However, forecasting river flow is challenging due to the complex and dynamic nature of hydrological data, which can be influenced by precipitation, reservoir operations, land-use changes, and other factors.

Understanding river flow dynamics is vital for operational management and the safety of communities in flood-prone areas. Advanced alert systems, powered by forecasting models, enable timely evacuations and preparedness actions that help preserve lives and minimize economic losses [1]. In the context of climate change, which has led to more frequent and intense extreme weather events, accurate river flow prediction has become even more critical [3].

Furthermore, the ecological integrity of river systems is closely related to the variability of water flow. Changes in flow regimes affect sediment transport, water quality, and habitat availability for aquatic organisms. Although forecast systems for estuarine areas have improved in recent years [4,5], predicting incoming river flows remains a significant challenge [4]. Deep learning models have gained prominence in time-series forecasting, including river flow prediction, due to their ability to capture complex patterns and temporal dependencies [1,5]. Recent comparative studies have shown that model performance can vary according to the network structure, hyperparameters, and training

methodologies used [1,6,7]. Consequently, selecting and optimizing the appropriate model is essential based on the specific forecasting objectives and the available data. Sensor monitoring networks play a central role in the establishment of AI river flow models. The conceptualization and implementation of these networks is key to gathering sufficient data for reliable AI modeling. In Portugal, a very comprehensive sensor network has been deployed for 40 years, denoted SNIRH, covering multiple types of sensors and sensor variables (river flow, water level, precipitation, and dam discharges). This network provides the adequate context for developing AI models. Furthermore, the data from many sensors is available in real time, enabling the use of trained and validated models for predictive purposes. The proposed work explores the richness of the data from this sensor network and illustrates how it can contribute to improving the quality of forecasted river flows.

Given the complexities of river flow forecasting and the wide range of ML and DL approaches, we pose the following research question:

"Can the systematic deployment of MLP models—optimized through extensive hyperparameter tuning and evaluated in comparison with alternative machine learning(ML) techniques—provide reliable and robust short-term river flow forecasts in dam-regulated rivers, under both normal and raised flow conditions?"

Our hypothesis is that MLP-based forecasting, when carefully compared and optimized alongside models such as LSTM, SVM, ELM, and RF, will generally yield robust and reliable river flow predictions. We expect that MLPs will demonstrate competitive performance under normal flow conditions and produce reasonably accurate results in more challenging high-flow scenarios. Although we do not claim that MLPs will always outperform all alternatives in extreme events, we anticipate that they offer a practical balance between forecasting ability and computational efficiency.

In this paper, we focus on applying MLPs for river flow forecasting and evaluate their efficacy relative to other ML and DL methods. The models considered include MLPs, LSTMs, SVMs, RFs, and ELMs. This initial exploration provides insights into the strengths and limitations of each approach and supports our decision to focus on MLPs for an in-depth analysis and for establishing forecast services that integrate with estuarine predictions.

Based on our findings, we define two specific scenarios for the Tejo River and one for the Mondego River, targeting the prediction of river discharge for the next three days using MLPs. These models integrate data from downstream dams including river discharge, reservoir outflows, reservoir levels, and precipitation—to improve prediction precision. Our objective is to demonstrate the applicability and robustness of MLPs in diverse hydrological and hydraulic contexts.

The research aims to develop and refine AI models that function as remote services, providing river flow forecasts for the current day and the following two days. These models combine advanced data techniques with reliable hydrological and meteorological data to improve water management decisions.

This study is part of the CONNECT project, within the Coastal Downscaling Program of the Copernicus Marine Service, and the ATTRACT project (Digital Innovation Hub for Artificial Intelligence and High-Performance Computing). Both projects focus on developing and implementing AI-based computational services to facilitate industry uptake of AI-driven products. The remainder of the paper is organized as follows. Section 2 reviews the relevant literature and previous studies. Section 3 outlines the case studies and methodology. Section 4 provides a concise formulation of the fundamental concepts and mathematical foundations of the ML and DL models employed in this study, including MLP, LSTM, SVM, ELM, and RF . Section 5 presents the application of the methodology and compares the performance of different models. Finally, Section 6 discusses the main findings and suggests directions for future research.

2. Related Work

River flow forecasting has received significant attention in the last decades due to its critical role in water resource management, flood mitigation, and environmental conservation [8]. The complexity of hydrological systems requires the development and application of advanced predictive models. ML

and deep learning (DL) models have emerged as prominent tools in this domain, offering enhanced capabilities to capture the intricate patterns inherent in river flow data [9]. However, each method has its own set of advantages and limitations, and a comprehensive understanding of them is essential to advance the field.

2.1. ML and DL Models for River Flow Forecasting

A diverse array of ML and DL models have been used for river flow forecasting, each with unique strengths and applications. Below we briefly review these methods and critically analyze their advantages and disadvantages for time-series forecasting :

- Autoregressive Integrated Moving Average (ARIMA)
  ARIMA models are traditional time series forecasting tools that capture temporal trends, seasonality, and time-dependent patterns in data. They are particularly effective in analyzing historical river flow data and providing baseline forecasts [10,11]. Despite their interpretability, ARIMA models often struggle with non-stationary data and the nonlinear patterns typical of hydrological processes. [8].

- Linear Regression (LR)
  LR models the relationship between river flow and external factors such as precipitation and temperature through a linear equation. Although simple and interpretable, LR is limited in its ability to capture nonlinear interactions and complex dependencies that often characterize river flow processes [12].

- Multilayer Perceptron (MLP)
  MLPs are feedforward neural networks capable of capturing nonlinear relationships in data. They are well suited for short-term river flow predictions, especially when high temporal resolution data is available. While computationally efficient and flexible, MLPs require extensive hyperparameter tuning and their performance is sensitive to the quality of input data. Furthermore, the default optimization algorithm (gradient descent) commonly used in MLPs often converges to local minima, particularly when handling highly stochastic time series data like river streamflow. This limitation, along with the risk of overfitting, can lead to inaccurate predictions, thus necessitating more advanced optimization strategies [13].

- Support Vector Machines (SVMs)
  SVMs are supervised learning models that predict continuous outcomes by finding the optimal hyperplane that separates data. They effectively model complex nonlinear relationships using kernel functions. In hydrological applications like river flow forecasting, this sensitivity may result in unreliable forecasts when the data display significant variations or noise, thereby decreasing overall forecasting accuracy [14].

- Random Forests (RFs)
  RFs are ensemble learning methods that construct multiple decision trees and merge their results to improve predictive accuracy and control overfitting [15]. As noted in [14], RF-based approaches sometimes underestimate extreme values in longer forecast horizons (e.g., 2- and 3-hour projections), thereby diminishing their ability to accurately predict severe hydrological phenomena.

- Extreme Learning Machines (ELMs)
  ELMs are single-hidden-layer feedforward neural networks with randomly assigned weights and biases, offering rapid training speeds and good generalization performance. Sometimes, ELMs face accuracy limitations when modeling complex and highly dynamic hydrological scenarios, despite their efficiency. Additional disadvantages include their dependence on random initialization, which can lead to inconsistent performance across experiments, and the need for a large number of hidden neurons to achieve competitive accuracy, which increases both computational complexity and the risk of overfitting [16].

- Recurrent Neural Networks (RNNs)
  RNNs are designed to handle sequential data by maintaining a memory of previous inputs, making them effective for modeling dynamic systems such as river flow. However, standard RNNs often face challenges such as vanishing gradients that hinder their ability to learn long-term dependencies. Furthermore, they can suffer from instability issues such as gradient explosion, require fine-tuning of learning rates and other meta-parameters, and generally struggle to capture long-term dynamics in complex hydrological time series [17].

- Long Short-Term Memory (LSTM)
  LSTMs, an advanced RNN architecture, overcome the vanishing gradient problem and capture long-term dependencies in time-series data [18]. However, the complexity and computational requirements of LSTMs, as highlighted by Rahimzad et al. (2021), can become significant challenges, especially with large datasets[19].

- Convolutional Neural Networks (CNNs)
  Originally developed for spatial data, CNNs have been adapted for time-series forecasting by capturing local patterns in hydrological data, such as precipitation and topography Because CNNs are inherently designed to extract spatial features, they can struggle with modeling long-term temporal dependencies in hydrological processes and may require large amounts of training data and careful kernel size design to effectively capture the sequential nature of the data[20].

- Gated Recurrent Unit (GRU)
  GRUs are streamlined versions of LSTMs with fewer parameters, resulting in faster training while still effectively capturing temporal relationships. They may not capture the fine temporal dynamics as well as LSTMs in all cases, but they offer a good compromise between complexity and performance. Furthermore, GRUs may sometimes fall short when modeling very complex or long-term dependencies in hydrological time series compared to more sophisticated architectures [21].

- Positive and Negative Perceptron (PNP)
  PNPs incorporate both positive and negative contributions within their architecture, aiming to capture diverse hydrological characteristics more effectively. Being relatively new, further research is needed to establish their stability and reliability in river flow prediction. Furthermore, their innovative structure may require more complex tuning and extensive validation to ensure robustness under different hydrological conditions, and their relative performance against traditional models remains to be comprehensively evaluated. [22].

- Attention-Based Neural Networks (AttNet)
  Recent studies have shown that attention-based models can significantly enhance streamflow forecasting by focusing on the most relevant temporal features [23,24]. Despite promising results, the increased complexity and computational demands of these models can be challenging in operational settings. Moreover, as highlighted by Liu et al. (2024) and Lee et al. (2024), these models often require extensive data segmentation, hyperparameter tuning, and significant computational resources to effectively manage and interpret complex hydrological data, which can impede their real-time application.

- Hybrid Models
  Hybrid models integrate multiple methodologies—such as combining wavelet transforms with ML algorithms—to capture both linear and nonlinear patterns in river flow data. Hybrid models, by combining multiple methods such as wavelet transform and ML algorithms, demonstrate more advanced forecasting capabilities by exploiting both linear and nonlinear components of hydrological datasets [25,26].

Recent literature emphasizes that a deep understanding of the trade-offs between these methods is crucial. Comprehensive evaluations that critically analyze the strengths, weaknesses and suitability of these models for specific forecasting tasks help inform the selection of the most appropriate model for a given hydrological challenge [27,28].

2.2. Research Overview

In our study, we identified a select group of ML and DL models that are particularly well suited to our case studies in river flow forecasting. This group includes LSTM networks, ELMs, RFs, SVMs, and MLPs. LSTM networks have emerged as a dominant approach in hydrological modeling due to their ability to effectively capture temporal dependencies and nonlinear patterns in time-series data. For example, Rahimzad et al.[19] found that LSTM networks consistently outperformed LR, MLP, and SVM in river flow prediction. Building on this, Bakhshi Ostadkalayeh et al.[29] enhanced LSTM models by integrating Kalman filtering to significantly reduce prediction errors. Similarly, Ho et al.[30] applied multi-step-ahead LSTM models to improve sluice gate operations in Vietnam, while Cho and Kim[31] merged LSTM with the WRF-Hydro model for improved streamflow predictions. Additional studies by Xie et al.[32], Xiang and Demir[33], and Ni et al.[26] further highlight the potential of hybrid LSTM-based approaches in capturing seasonal variations and extreme hydrological events. Nguyen et al.[34] developed a deep neural network with LSTM layers to predict flow in the Mekong River Basin. Their results showed that the model could accurately capture seasonal patterns and abrupt changes in flow. In summary, the LSTM components were key to modeling the complex temporal dependencies in the Mekong flow regime. The results indicated that, compared with traditional approaches, the LSTM-based model achieved robust and reliable predictions under different hydrological conditions, demonstrating its strong potential for large-scale river basin applications. In addition, Hunt et al.[35] demonstrated that integrating LSTM networks into streamflow forecasting frameworks can significantly enhance the accuracy of predictions over the western United States.

ELMs have gained traction in hydrological forecasting for their rapid training capabilities and efficiency in handling nonlinear relationships [16]. Furthermore, Bărbulescu and Liu [36] compared various AI methods for river water discharge forecasting and found that, although advanced DL approaches often provide high accuracy, ELMs offer a competitive alternative due to their significantly faster training times and lower computational demands, making them suitable for real-time forecasting applications. RFs, introduced by Breiman [15], have also been shown to outperform other algorithms in terms of accuracy and robustness under varying hydrological conditions [37]. Additionally, Islam et al.[38] investigated the use of random forest regression with remotely sensed data to predict streamflow in a snowmelt-dominated watershed. Their study showed that a customized RF approach outperformed the physically based SWAT model—especially when trained over long periods—with snow depth and minimum temperature as the most critical predictors. SVMs have been widely used in hydrological modeling because of their proficiency in capturing complex nonlinear relationships, although they can be computationally intensive for large-scale applications [14]. Notably, Mahmood et al.[39] investigated reservoir inflow forecasting for the Haditha Reservoir in western Iraq using SVM. Their study showed that SVM can be effectively applied in data-poor conditions and achieved competitive performance compared to neural network models, emphasizing that SVM remains a viable option for flood and inflow forecasting in complex hydrological settings.

Dibike and Solomatine [40] demonstrated the effectiveness of ANNs, especially MLPs, in the capture of complex hydrological patterns. Their study compared MLPs with a conceptual rainfall–runoff model and found that the ANN approach performed slightly better than traditional methods in predicting river flow. Brandão et al. [41] investigated flood forecasting in an ungauged basin using the Paranaíba River as a case study. This work demonstrates that artificial neural networks, specifically MLPs, can effectively capture the nonlinear dynamics of flood events in data-poor environments, offering a promising tool for early flood warnings.

Finally, MLPs remain a fundamental component in river flow forecasting due to their flexibility and capacity to model nonlinear patterns, especially when combined with advanced feature engineering techniques like Principal Component Analysis (PCA) [13].

3. Methodology

This section describes the case studies and the methodology for forecasting river flow (Figure 1) for the next one, two, and three days. The workflow comprises three main steps: preprocessing, model development, and model validation and forecasting. Each step is tailored to the data available for each basin, with the selection of data stations based on expert knowledge of river dynamics and basin-specific characteristics.
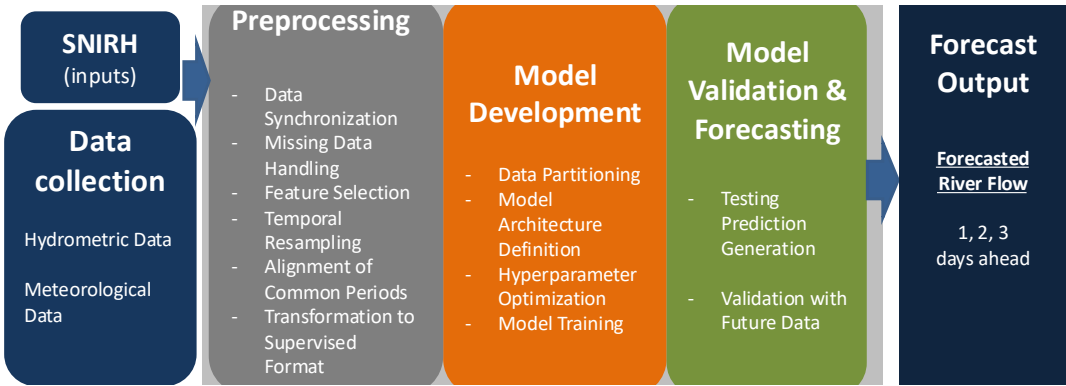

Figure 1. Methodology architecture diagram

3.1. Case Studies: Tejo and Mondego Rivers

We focus on forecasting the daily river flow for two important Portuguese rivers: the Tejo (Tagus) and the Mondego. Data for both rivers were sourced from Portugal's national water resource information system, the *Sistema Nacional de Informação de Recursos Hídricos* (SNIRH) [42], which provides comprehensive hydrological and meteorological data essential for accurate forecasting models.

Leveraging the extensive dataset provided by SNIRH—widely recognized for its effective monitoring of aquatic environments [42]—establishes a strong basis for creating accurate forecasting models.

3.1.1. Tejo River

The Tagus basin (Figure 2) is one of the largest in the Iberian Peninsula. The Tejo River spans 1,007 km and covers a basin area of 80,626 km^2 (24,845 km^2 in Portugal). Its flow regime is primarily controlled by several dams located on both the main river and its tributaries [43].

Flood events in the Tejo River are frequent, with the 1979 floods being particularly severe. Reliable river flow predictors are therefore essential. Accurate forecasts are also important for the Tagus estuary (developed in the CONNECT project), where the freshwater/saltwater balance critically influences water quality [44].

For the Tejo River, two datasets were used to forecast the daily river flow at Almourol hydrometric station for the period from October 1, 1984, to September 26, 2023:

3.1.2. Mondego River

The Mondego River basin (Figure 3) covers 6,645 km^2, making it the second largest basin entirely within Portugal. Stretching 234 km, the Mondego River originates in mountainous regions, flows into a wide alluvial flood plain, and ultimately empties into the Atlantic Ocean. Its flow is regulated by dams—Aguieira, Raiva, Fronhas, and Açude-Ponte Coimbra—constructed during the 1980s.

Flooding is a major concern in the Mondego basin [45]. Infrastructure such as lateral dikes and flood-control structures (e.g., at the Açude-Ponte in Coimbra) have been constructed to mitigate flood impacts. Accurate river flow predictions are critical for both flood management and for estimating freshwater intake for the Mondego estuary forecast, where freshwater flows significantly affect the salt balance during heavy precipitation events.

For the Mondego River, only stations with long daily records and real-time data were selected (excluding precipitation data). The dataset consists of daily discharges from the Aguieira, Raiva, Fronhas, and Açude-Ponte Coimbra dams, covering the period from November 4, 1997, to March 9, 2024. The AI model was developed to predict daily discharges at the Açude Ponte Coimbra (12G/01AE) station.

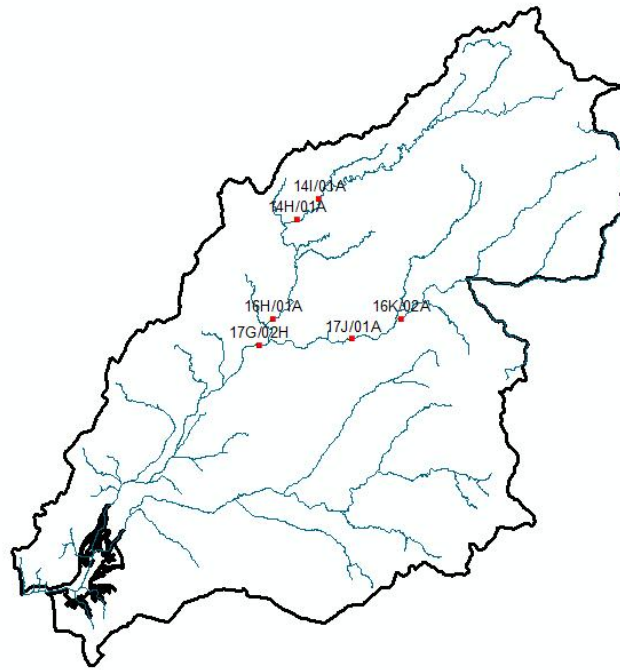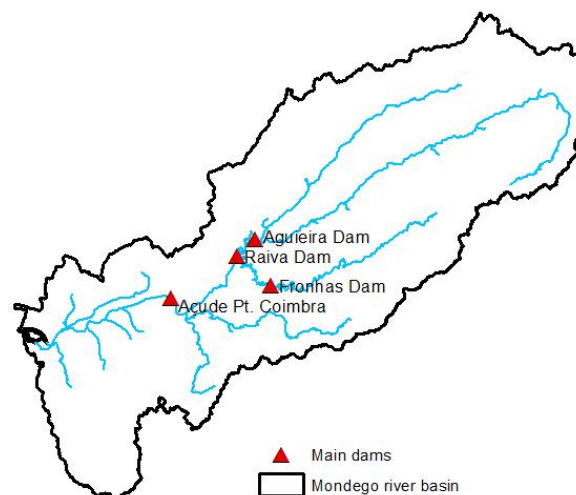**Figure 2.** Locations of hydrometric measuring stations in the (a) Tejo

Figure 3. Locations of hydrometric measuring stations in the (b) Mondego basins

3.2. AI Model Construction

The process of developing robust *artificial intelligence (AI)* river flow forecasting models consists of multiple essential stages, each specifically tailored to guarantee the precision and dependability of

the predictions. This section offers a comprehensive explanation of the steps required, beginning with data preprocessing, then moving on to model training, and concluding with the forecasting phase.

3.2.1. Data Collection

Data for this investigation were obtained from national hydrometric and meteorological monitoring networks to guarantee extensive coverage and dependability. The datasets included hydrological variables, including river discharge and effluent flow, and meteorological data, such as precipitation and temperature, pertinent to river flow dynamics. Two scenarios were examined: one concentrated on hydrological data from dam-regulated discharges, while the other included meteorological inputs from nearby regions. These scenarios were created to investigate the prediction capacity of AI algorithms to display river flows in certain target areas.

3.2.2. Preprocessing Steps

The following preprocessing steps were taken to ensure data quality, consistency, and integrity:

- Data Synchronization: Initially, datasets from various stations, including hydrometric and meteorological stations monitored by the Sistema Nacional de Informação de Recursos Hídricos (SNIRH), are acquired and formatted into a consistent structure. This involves parsing date-time information, normalizing measurement units, and synchronizing records from different stations to establish a unified timeline. Normally, the dataset is loaded from a *comma-separated values (CSV)* file, downloaded from the SNIRH website, and the date column is converted to a datetime format. The date column is then set as the index of the DataFrame to facilitate time series analysis.

- Missing Data Handling: Ensuring the completeness of the data is essential to preserve the accuracy and reliability of the dataset. To properly handle missing values, several techniques are employed, including linear interpolation and forward and backward filling. However, due to the nature of river flow data, filling in missing values can introduce inaccuracies. Therefore, we create a set of functions to promote these assessments.

- Feature Selection: It is crucial to identify and select the key variables that significantly influence the predictive model's performance. These variables include historical river flow discharge measurements, meteorological data (such as precipitation), and other relevant factors, such as dam discharge rates. For this study, we promoted several selections based on the performance of the constructed model.

- Temporal Resampling: To achieve consistency, the data's resolution is standardized through temporal resampling. The dataset is resampled to a daily frequency to ensure uniform time intervals. This step aggregates the data and fills any missing dates with interpolated values.

- Alignment of Common Periods: It is crucial to synchronize datasets from several stations so they overlap within the same time periods. This guarantees that models are trained on datasets encompassing all pertinent characteristics within the same time frame.

  Due to the presence of significant missing values, we focus on combining periods with minimal missing data to ensure the integrity of the dataset. The `get_common_periods_sections` function is used to identify periods with a maximum of 10 missing values. This approach helps maintain the validity of the river flow data while ensuring enough data is available for model training. The steps are shown in Algorithm 1, which includes combining datasets, figuring out missing values, and selecting acceptable time segments based on certain criteria.

Algorithm 1 Identify Common Periods with Minimal Missing Values and Fill Missing Data

---

Require: – `dataframes`: List of DataFrames from different stations
    – `max_missing`: Maximum allowed missing values per day (e.g., 10)
    – `min_required_period`: Minimum number of consecutive days required
Ensure: – `common_periods`: List of start and end dates with minimal missing data
    – `filled_data`: DataFrames with missing values filled by interpolation
 1: Initialize `common_periods` as an empty list
 2: Initialize `filled_data` as an empty list
 3: Merge all `dataframes` on the datetime index using an outer join
 4: Calculate the total number of missing values per day
 5: Create a boolean mask where missing values $\leq$ `max_missing`
 6: Find continuous True segments in the mask
 7: **for** each continuous segment **do**
 8:   **if** length of segment $\geq$ `min_required_period` **then**
 9:     Append (start_date, end_date) to `common_periods`
10:     Extract data for the segment
11:     Fill missing values in the segment using interpolation
12:     Append filled segment to `filled_data`
13:   **end if**
14: **end for**
15: **return** `common_periods, filled_data`

- Transformation to Supervised Learning Format: Time series data is converted into a format suitable for supervised learning. The `series_to_supervised` function transforms the time series data into a supervised learning problem by creating lagged versions of the input features. This transformation allows the model to learn temporal dependencies in the data. The function creates input sequences of length n_{in} and output sequences of length n_{out}. To generate input-output pairings, Algorithm 2 methodically shifts the data. The future values $(t+1, \ldots, t+n_{out})$ function as targets, and the lag features $(t-n_{in}, \ldots, t)$ as model inputs.

Algorithm 2 Transform Time Series to Supervised Learning Format

---

Require: – `data`: Time series data as a DataFrame
    – `n_in`: Number of lag observations as input (e.g., 3)
    – `n_out`: Number of observations as output (e.g., 1)
Ensure: `supervised_data`: Transformed DataFrame suitable for supervised learning
 1: Initialize `cols` as an empty list
 2: **for** i = -n_in **to** 0 **do**
 3:   `cols.append(data.shift(i))`
 4: **end for**
 5: **for** j = 1 **to** n_out **do**
 6:   `cols.append(data.shift(-j))`
 7: **end for**
 8: Concatenate `cols` along the column axis
 9: Drop all rows with NaN values
10: Rename columns appropriately (e.g., var(t-n), ..., var(t), var(t+1), ...)
11: **return** `supervised_data`

3.2.3. Model Development

The model training stage involves splitting the data into training and testing sets, defining the model architecture, and training the model. The key steps are as follows:

- Data Partitioning: The combined supervised data is split into training and testing sets using an 80–20 ratio. The training set is used to train the model, while the testing set is used to evaluate its performance.

- Model Architecture Definition: The structure of the ML models is determined according to the particular needs of the forecasting task. This involves choosing the appropriate number of layers, neurons per layer, activation functions, and other architectural characteristics for models such as *LSTM*, *MLPs*, *ELMs*, *RFs*, and *SVMs*. We mainly used the Keras library to define the model. Multiple instances of these models were trained and selected based on performance. For example, for MLPs, we used multiple dense layers with ReLU activation functions and dropout layers to avoid overfitting. The input dimension of the initial layer is set to the number of features in the training data, while the output dimension is set to the number of forecasting steps (three days).
- Hyperparameter Optimization: To improve model performance, hyperparameters were fine-tuned using grid search. The goal of this stage was to determine the best combination of hyperparameters that maximizes the predicted accuracy of the model. Depending on the model configuration, we vary the number of neurons per layer, the number of epochs, L2 regularization values, dropout rates, batch sizes, optimizers, and early stopping patience.
- Model Training: Our models are trained using the Adam optimizer and mean squared error (MSE) as the loss function. Early stopping is used to monitor the validation loss and prevent overfitting. The model is trained for a specified number of epochs, and the best model weights are saved based on the validation loss.

3.2.4. Model Validation and Forecasting

The forecasting stage uses the trained models to generate predictions for three days. The key steps are as follows:

- Prediction Generation on Test Data: The trained models generate predictions based on the test data. To evaluate model performance, the root mean squared error (RMSE) is calculated for each forecasting step (today, tomorrow, and the day after tomorrow).
- Validation with Future Data: The models are validated on a separate validation set containing recent data. For example, we currently use the entire 2023 dataset. The validation data is preprocessed and transformed in the same manner as the training data. Each model generates predictions for the validation period, and the RMSE is calculated for each forecasting step.

4. Theoretical Background

The fundamental concepts and essential equations for the *ML* and *DL* models used in this study are explained in this section.

4.1. Multilayer Perceptron

A multilayer perceptron [46,47] is a feed-forward neural network composed of an input layer, one or more hidden layers, and an output layer. Each neuron in a hidden layer computes:

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{1}$$

where $\mathbf{x}$ is the input vector, $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector, and $f(\cdot)$ is a nonlinear activation function (e.g., ReLU, sigmoid, or tanh). The output is obtained by applying a similar operation at the final layer.

4.2. Long Short-Term Memory Networks

A long short-term memory network [48] is a specialized type of *recurrent neural network (RNN)* designed to capture long-term temporal dependencies in sequential data. The internal gating mechanisms in an LSTM cell are defined as:

$$i_t = \sigma\big(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\big), \qquad\qquad f_t = \sigma\big(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f\big), \tag{2}$$

$$o_t = \sigma\big(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\big), \qquad\qquad \tilde{\mathbf{c}}_t = \tanh\big(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c\big), \tag{3}$$

$$\mathbf{c}_t = f_t \odot \mathbf{c}_{t-1} + i_t \odot \tilde{\mathbf{c}}_t, \qquad\qquad \mathbf{h}_t = o_t \odot \tanh\big(\mathbf{c}_t\big), \tag{4}$$

where i_t, f_t, o_t denote the input, forget, and output gates, respectively; $\mathbf{c}_t$ is the cell state; $\mathbf{h}_t$ is the hidden state; and $\odot$ is element-wise multiplication. The function $\sigma(\cdot)$ is the sigmoid activation, and $\tanh(\cdot)$ is the hyperbolic tangent.

4.3. Support Vector Machine

Support vector regression [49] aims to find a function $f(\mathbf{x})$ that deviates from the actual target values by at most ϵ. This is formulated as:

$$\min_{\mathbf{w},b,\xi,\xi^*} \quad \frac{1}{2}\|\mathbf{w}\|^2 \;+\; C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{5}$$

$$\text{subject to} \quad \begin{cases} y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \;\le\; \epsilon + \xi_i, \\ (\mathbf{w}^\top \mathbf{x}_i + b) - y_i \;\le\; \epsilon + \xi_i^*, \\ \xi_i, \; \xi_i^* \;\ge\; 0, \end{cases} \tag{6}$$

where C is a regularization parameter, and $\xi_i, \; \xi_i^*$ are slack variables controlling the allowed error.

4.4. Extreme Learning Machine

An extreme learning machine [16] is a single-hidden-layer feed-forward network. The hidden layer weights and biases are randomly initialized and remain fixed, while the output weights are computed analytically via:

$$\boldsymbol{\beta} = \mathbf{H}^\dagger \, \mathbf{T}, \tag{7}$$

where $\mathbf{H}$ is the hidden layer output matrix, $\mathbf{T}$ is the target matrix, and $\mathbf{H}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{H}$.

4.5. Random Forest

A random forest [50] is an ensemble of decision trees. For a given input $\mathbf{x}$, each tree t outputs a prediction $h_t(\mathbf{x})$. The final prediction is the average of all tree outputs:

$$\hat{y}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} h_t(\mathbf{x}), \tag{8}$$

where T is the total number of trees.

5. Application to the Case Studies

This section applies the previous methodology to two scenarios involving the Tejo and Mondego rivers. We evaluated the performance of various ML models, including LSTM networks, MLP, SVM, ELM, and RF. Through systematic hyperparameter tuning and performance comparison, we identify the most effective model for river flow forecasting in each case study.

5.1. Comparison of Models Performance for Tejo River and Selection of MLP

The comparative performance of different ML models in forecasting daily river discharge for the Tejo River is presented in Table 6. The evaluation focuses on the RMSE for predictions for today and tomorrow.

To perform a thorough analysis, we designed several experiments using different input stations within the Tejo River, with the Almourol station as the target for discharge predictions. The selected input stations include:

- **0. Castelo de Bode Average daily dam outflow discharge (m³/s)**
- **1. Castelo de Bode Reservoir (m)**
- **2. FratelAvaerage daily dam outflow discharge (m³/s)**
- **3. Fratel Reservoir water level (m)**
- **4. Almourol Daily Average discharge (m³/s)**

For evaluation, we categorized the experiments into four scenarios:

- Scenarios **a**, **b**, and **c**: Validation period from **2022-08-07 to 2023-09-04**.
- Scenario **d**: Validation period from **2003-03-31 to 2004-11-07**.

The scenarios were chosen based on hydrological significance, data accessibility, forecasting goals, and model complexity, with a focus on critical hydrometric stations that have minimal data gaps and extensive datasets. We also considered scenarios with extreme conditions to assess model robustness under both typical and exceptional circumstances. First, we preprocessed all time series without missing values and combined them into a comprehensive dataset. We then trained MLP and LSTM models to evaluate their ability to capture temporal relationships. Based on their performance, the MLP was selected as the preferred model. Subsequently, the MLP was compared with other ML models (i.e., ELM, SVM, and RF). While ELM and SVM did not achieve adequate prediction performance, RF demonstrated strong accuracy for 1-day forecasts but poorer performance for 2-day predictions. Overall, the MLP consistently achieved a lower RMSE across most scenarios and forecasting horizons using systematic hyperparameter tuning (including grid search).

Analysis of Performance Values:

Examining Table 6 reveals several key insights:

- **Scenario a** (Input: [0,2] → [4], 2-day forecast):
    - The MLP configurations yielded RMSE values of 162.65 (today) and 227.02 (tomorrow) in one configuration. In comparison, the LSTM models produced slightly higher errors (168.70 and 239.71, respectively).
    - SVM reported RMSE values of 367.23 and 367.31 in one configuration and 203.01 and 231.81 in another, indicating sensitivity to hyperparameter selection.
    - Although RF achieved an RMSE of 155.71 for the 1-day forecast, its error increased to 264.51 for the 2-day prediction.
    - ELM results were generally higher, with one configuration reporting 326.42 and 334.82, and another with 307.51 and 308.27.

- **Scenario b** (Input: [0,2,3] → [4], 2-day forecast):
    - The MLP produced RMSE values of 152.71 and 213.88 in one configuration and 149.14 and 227.87 in another, suggesting that including an additional input (from station 3) improved the performance.
    - The LSTM model yielded RMSE values of 141.85 and 216.48, while the SVM and RF achieved 198.12, 230.45 and 148.79, 261.46 respectively.
    - ELM reported values of 218.08 and 255.18.

- **Scenario c** (Input: [0,1,2,3,4] → [4], 2-day forecast):

- – The MLP achieved RMSE values as low as 136.71 and 206.49 in one configuration and 141.39 and 217.94 in another.
- – LSTM values were 136.02 and 212.92, while SVM produced 153.17 and 200.09.
- – RF and ELM in this scenario recorded RMSE values of 149.53, 257.72 and 103.82, 179.47 (with the latter configuration for ELM highlighting the potential for lower error in one output), respectively.

- **Scenario d** (Input: [0,2,4] → [4], 1-day forecast):
  - – The MLP reported an RMSE of 104.53 for the 1-day forecast, which is slightly better than SVM's 105.75 and notably lower than LSTM's 121.51 and RF's 119.11.
  - – ELM, however, showed a higher RMSE of 223.36 (with one additional configuration at 356.93 in the LSTM column).

These results indicate that each model's performance is closely tied to its hyperparameter configuration and the specific input combination used. While some models such as RF or SVM may exhibit competitive performance under certain conditions, the systematic tuning process confirms that the MLP approach produces competitive error metrics across the majority of the scenarios—especially when evaluated across varying forecast horizons and input combinations. The tuning process not only helped optimize model parameters for the given dataset but also provided insights into the relative strengths and limitations of each approach in capturing the nonlinear dynamics of river flow.

In summary, the comparative analysis shows that—although several models can yield competitive results under specific settings—the performance of the MLP is promising for our dataset. This evaluation supports the further investigation of the MLP approach for river flow forecasting in the Tejo basin, while acknowledging that model performance can vary with different configurations and input selections.

Table 1. LSTM Model Configurations for Use scenarios

Sc	Cfg.	LSTM Layers (Units, Act., Seq.)	Dropout	Output (Units, Act.), Optimizer, Batch, Epochs
a	1	96, ReLU, T → 96, ReLU, F → 96, ReLU, F	0.3 / - / 0.2	2, Softplus, Adam (lr=0.00929), 32, 50
b	1	128, ReLU, T → 128, ReLU, F → 96, ReLU, F	0.4 / 0.3 / -	2, Linear, Adam (lr=0.00691), 32, 50
	2	50, ReLU, F	-	2, Linear, Adam, 32, 100
c	1	64, ReLU, T → 128, ReLU, F → 40, ReLU, F	0.1 / - / 0.1	2, Linear, Adam (lr=0.00576), 32, 50
d	1	32, LSTM, T → 90, LSTM, F → 96, LSTM, F	0.1 / - / -	1, Linear, Adam (lr=0.00022), 32, 50
	2	50, ReLU, T → 30, ReLU, F	0.1	1, SELU, Adam, 16, 1000

Table 2. MLP Configurations for Use scenarios

Sc	Cfg.	Hidden Layers (Units, Act.)	L2 Reg	Output (Units, Act.), Optimizer, Batch, Epochs
a	1	[90, 90], ReLU	0.01	2, SELU, Adam (lr=0.001), 16, 100
b	1	[150, 40], ReLU	0.01	2, SELU, Adam (lr=0.001), 16, 13
	2	[150, 40], ReLU	0.01	2, SELU, Adam (lr=0.001), 16, 300
c	1	[90, 90], ReLU	0.01	2, SELU, Adam (lr=0.001), 16, 100
	2	64, ReLU → 90, ReLU	0.01	2, ReLU, Adam (lr=0.0003916), 16, 100
d	1	150, ReLU → 150, ReLU → 90, ReLU	0.01	1, ReLU, Adam (lr=0.0003097), 16, 100

Table 3. ELM Configurations for Use scenarios

Sc	Cfg.	Hidden Neurons	Activation
a	1	90	sigm
	2	150	sigm
b	1	200	tanh
c	1	50	tanh
d	1	90	sigm

Table 4. SVM Configurations for Use scenarios

Sc	Cfg.	C	Gamma	Epsilon
a	1	3	scale	0.02
	2	5	0.001	0.02
b	1	10	scale	0.02
c	1	100	scale	0.2
	2	100	scale	0.2
d	1	100	scale	0.5

Table 5. RF Model Configurations for scenarios

Sc	Cfg.	n_Estimators, max_Depth, min_Samples_Split
a	1	100, Default (None), Default (2)
b	1	100, Default (None), Default (2)
c	1	100, None (Day-1) → 10 (Day-2), 10
d	1	100, 30, 10

Table 6. Comparative Performance of ML and DL Models for River Flow Forecasting

Sc.	Input / Output	Days	MLP	LSTM	SVM	RF	ELM
a	$[0,2] \rightarrow [4]$	2	162.65, **227.02**	168.70, 239.71	367.23, 367.31 203.01, 231.81	155.71, 264.51	326.42, 334.82 307.51, 308.27
b	$[0,2,3] \rightarrow [4]$	2	152.71, **213.88** 149.14, 227.87	141.85, 216.48	198.12, 230.45	148.79, 261.46	218.08, 255.18
c	$[0,1,2,3,4] \rightarrow [4]$	2	136.71, 206.49 141.39, 217.94	136.02, 212.92	153.17, 200.09	149.53, 257.72	103.82, **179.47**
d	$[0,2,4] \rightarrow [4]$	1	**104.53**	121.51 356.93	105.75	119.11	223.36

Why MLP?

The MLP model regularly exceeds other ML models in river flow forecasting, as evidenced by its reduced RMSE values in most cases. This success is primarily due to its capacity to precisely model complex nonlinear relationships in the data, which is especially advantageous for forecasting sophisticated river flow dynamics. Moreover, MLP provides considerable computational efficiency relative to deep learning models such as LSTM (Long Short-Term Memory), requiring fewer computational resources while maintaining strong performance. This makes MLP a viable option for handling large datasets or when computing resources are limited. The adaptability and scalability of MLP architectures allow customization to suit various types and sizes of datasets, thus enhancing their relevance to diverse river flow forecasting applications [13,51]. Furthermore, previous studies have supported the effectiveness of MLP in forecasting time series, affirming its reliability in projecting river flow based on historical data [9,52,53]. These combined advantages position MLP as our study's leading model for river flow prediction, outperforming alternatives such as SVM, RF, and ELM.

- **Hyperparameter Optimization:** MLP requires fine-tuning of parameters such as the number of neurons in hidden layers, learning rate, and regularization coefficients. Identifying the optimal configuration can be time-consuming and necessitates extensive experimentation, particularly with large and complex datasets [27,54].
- **Overfitting:** MLP is susceptible to overfitting when the model complexity exceeds the available training data. Overfitting can lead to excellent performance on the training dataset but poor generalization to unseen data. Although regularization techniques such as L2 regularization and dropout can mitigate this issue, they require meticulous calibration to balance model complexity and performance [55].

Despite these limitations, the MLP model remains effective in predicting daily river discharge within the scope of this investigation. The dataset is relatively stable, and the primary challenge lies in identifying complex patterns in the time series data.

MLP was selected for its effectiveness in handling time series data. The architecture consisted of an input layer, two hidden layers, and an output layer designed to predict river discharge for three consecutive days: today, tomorrow, and the day after tomorrow. Hyperparameters were fine-tuned using grid search to optimize performance, and early stopping was implemented to prevent overfitting during training.

5.2. Model Configurations and Forecasting Results

Here, we describe the model configurations and assess the accuracy of forecasting river flows for the Mondego and Tejo Rivers. To enable different scenarios, the dataset was divided into periods for training, testing, and validation. Table 7 summarizes the dataset sizes for each phase, categorized by river and scenario.

Table 7 summarizes the size of the dataset for each phase.

Table 7. Training, Testing, and Validation Sets for Tejo and Mondego Rivers

River	scenario	Training Set	Testing Set	Validation Set
Tejo	A	8042 samples, 60 features	2011 samples, 60 features	350 samples, 60 features
	B	7120 samples, 100 features	1780 samples, 100 features	350 samples, 100 features
Mondego	A	3643 samples, 80 features	911 samples, 80 features	1873 samples, 80 features

Mondego River

For scenario A, three distinct MLP models were developed with varying hyperparameters. The RMSE values for each model in different forecasting horizons are presented in Table 8.

Table 8. Comparative RMSE of Different MLP Models for scenario A in Mondego River

Model	RMSE Today	RMSE Tomorrow	RMSE Day After
MLP1	26.79	**27.91**	**28.24**
MLP2	24.51	44.87	66.21
MLP3	34.01	45.54	54.04

Note: Validation period from 2024-01-01 to 2024-08-12.

MLP1 was identified as the best-performing model based on its consistent and lower RMSE values across the RMSE Tomorrow and RMSE Day After forecasting horizons. Consequently, it was selected for validation and further application in our scenario. MLP1 was configured with a hidden layer containing 50 neurons and trained for 50 epochs.

Tejo River

- **scenario A:** Two hidden layers with 90 neurons each, trained for 100 epochs.
- **scenario B:** Two hidden layers with 150 neurons in the first hidden layer and 40 neurons in the second hidden layer, trained for 300 epochs.

This configuration ensures that custom models are applied to each dataset, effectively capturing the unique influences of river flow, dam operations, and precipitation on discharge predictions.

5.3. Performance Metrics

To evaluate the accuracy of the forecasts, we used the root mean square error (RMSE) and bias, defined as follows ((see, e.g., [56]):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (P_i - O_i)^2}, \tag{9}$$

$$\text{Bias} = \frac{1}{n} \sum_{i=1}^{n} (P_i - O_i), \tag{10}$$

where P_i is the predicted value, O_i is the observed value, and n is the total number of observations.

- *RMSE* (Equation 9) quantifies the overall magnitude of prediction errors by squaring individual differences before averaging, thus placing a heavier penalty on larger errors. A lower RMSE indicates better agreement between predictions and observations.
- *Bias* (Equation 10) measures the systematic offset between the model and the observations. A positive bias means the model tends to overpredict, while a negative bias indicates underprediction.

5.4. Model Evaluation

The performance of the MLP models was assessed by comparing their predictions with the observed river flow discharge values for both rivers in different scenarios. A smaller RMSE and a bias closer to zero indicate better forecasting ability. Table 9 presents the RMSE and bias values for each scenario and forecast horizon.

Table 9. Evaluation of MLP Model Performance for Tejo and Mondego Rivers

River	Scenario	Forecast Horizon	RMSE (m^3/s)	Bias (m^3/s)
Tejo	A	1-Day (Today)	163.1	-22.5
	A	2-Day (Tomorrow)	212.1	-20.0
	A	3-Day (After Tomorrow)	228.4	-18.9
	B	1-Day (Today)	169.9	-5.6
	B	2-Day (Tomorrow)	215.1	-25.8
	B	3-Day (After Tomorrow)	232.6	-21.5
Mondego	A	1-Day (Today)	49.05	2.08
	A	2-Day (Tomorrow)	72.80	-0.76
	A	3-Day (After Tomorrow)	86.50	-3.62

Figure 4 presents a comparison between the measured and predicted values for the Tejo River in scenario A. Similarly, Figure 5 illustrates the performance for scenario B, while Figure 6 shows the results for the Mondego River in scenario A.
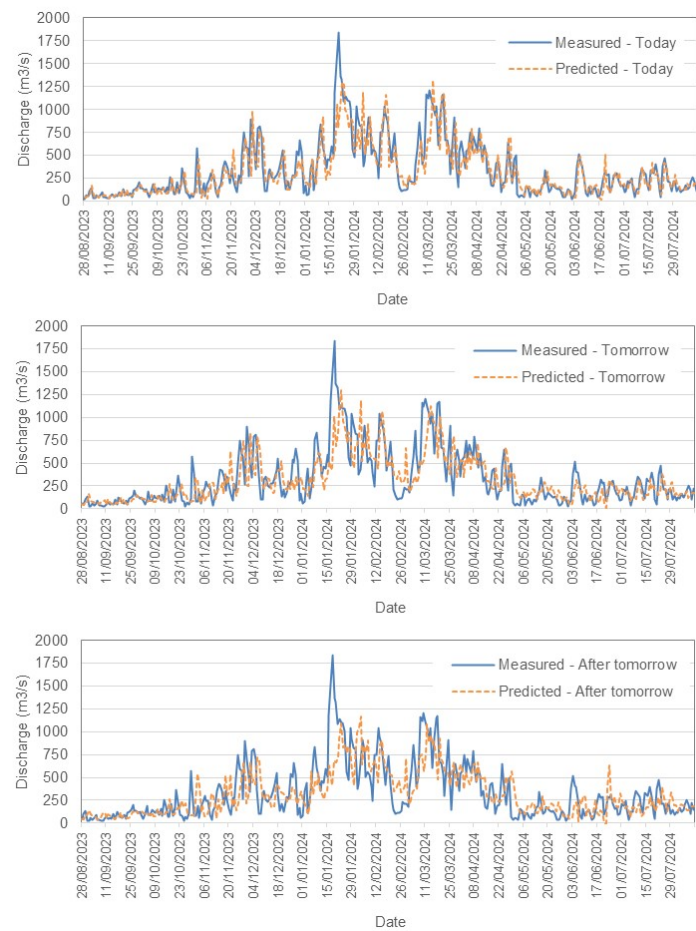
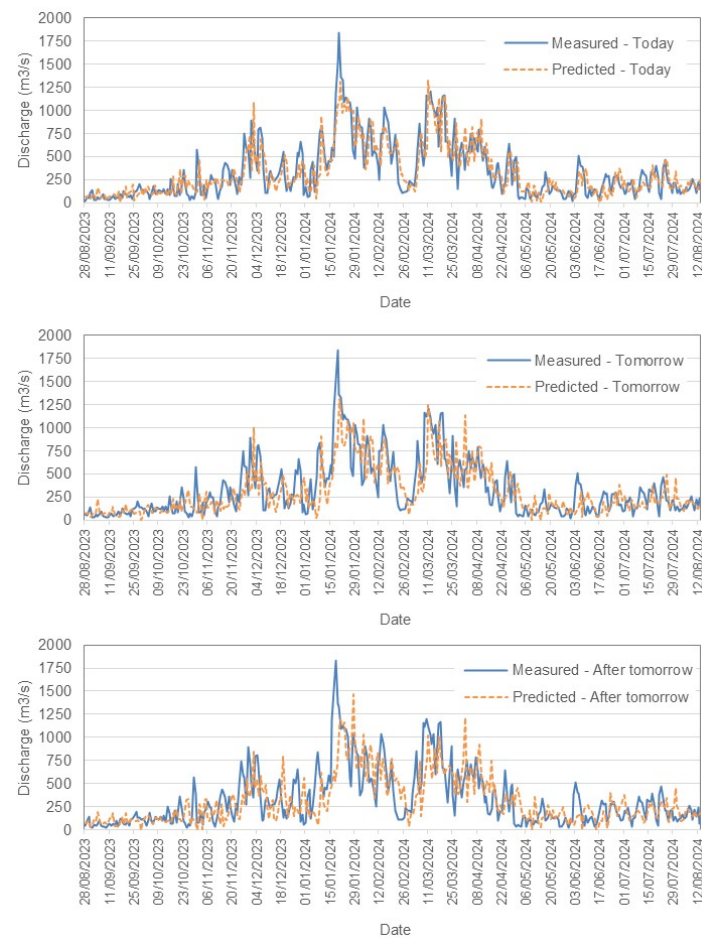**Figure 4.** Comparison between measured and predicted values Tejo River, scenario A

Figure 5. Comparison between measured and predicted values Tejo
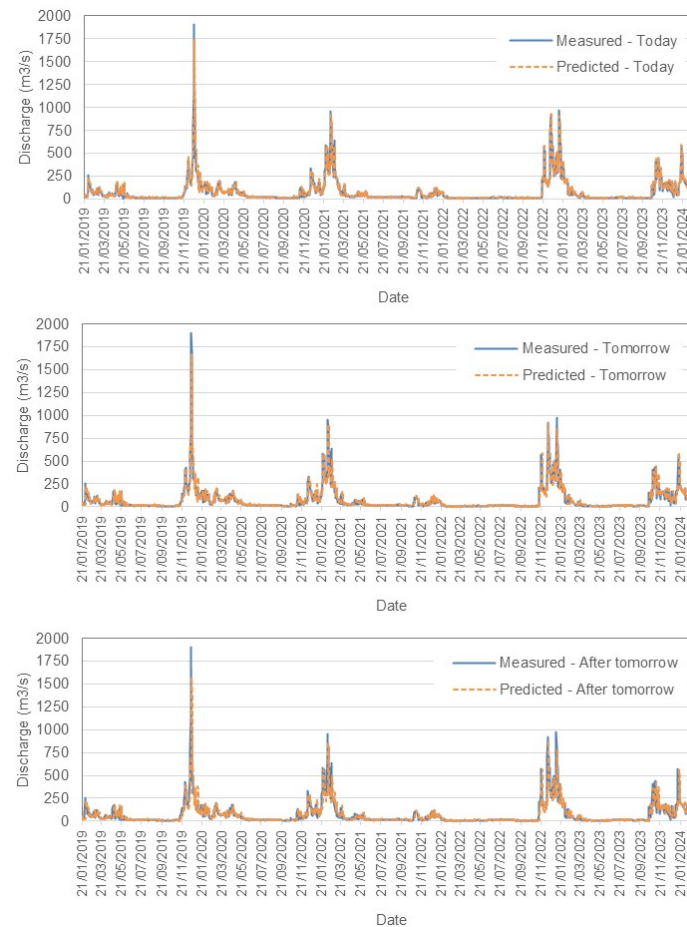River, scenario B

Figure 6. Comparison between measured and predicted values —
Mondego River, scenario A

– **Tejo River Results**

For the Tejo River, the model accurately captures short-term trends, especially for today's flow, with acceptable accuracy. However, as predictions extend further into the future, slight increases in RMSE reflect growing uncertainty. The model effectively tracks peaks and troughs but shows some under and over-predictions as the forecast horizon grows. Both scenarios demonstrate reliable short-term forecasting, and scenario B incorporates rainfall data to provide an alternative view of the influence of precipitation.

– **Mondego River Results**

In contrast, the Mondego River models exhibit lower RMSE values, with 49.05 m^3/s for 1-day forecasts (scenario A). This indicates a higher predictive accuracy compared to the Tejo River. Similar trends of increasing RMSE are observed with longer forecast horizons, although the absolute errors remain smaller. The bias values are closer to zero, suggesting more balanced predictions.

6. Conclusion and Future Work

This study explored the rich datasets from a comprehensive multi-sensor river sensor network to create MLP-based forecasting to predict daily river discharges under multiple hydraulic conditions. The work was demonstrated for two major Portuguese basins, and focused on very high river flow periods. The selection of MLPs was based on a comparative analysis of deep learning methods applied to the most complex of the two use cases. Then several models with MLPs were built to achieve adequate prediction accuracy. Although MLPs adequately record general discharge patterns, forecasting peak flows from intense rainfall or abrupt dam releases remains difficult. These infrequent

and rapidly occurring phenomena bring about ambiguities that complicate accurate predictions. However, the developed models continue to provide significant early warnings, making them an invaluable asset for proactive flood control. In addition, they compare favorably with traditional persistence approaches based on the last measured river flow.

The general findings for the Tejo and Mondego rivers are both satisfactory and justifiable. Despite increasing prediction errors during high discharge events, MLP models consistently provide prompt notifications for elevated water levels. They surpass or equal other ML methodologies under standard flow circumstances, demonstrating their dependability. In addition, the models accommodate various inputs, including historical flows, precipitation data, and dam operations, showing adaptability to the different circumstances in the two basins.

Enhancing peak-flow predictions requires training techniques centered on flood occurrences, such as oversampling peak-flow data or assigning more weight to mistakes in high-discharge scenarios. Furthermore, the use of near-real-time precipitation data and detailed dam operational information should facilitate the detection of rapid alterations often seen before floods.

Subsequently, we want to evaluate our models over a broader spectrum of flood scenarios to ensure the inclusion of diverse meteorological and hydrological trends. Collaboration with water management authorities and civil protection organizations is essential to synchronize the results of the models with daily activities in flood-prone regions. Focusing on these peak flow occurrences and integrating more comprehensive information, our goal is to improve MLP-based forecasting into a reliable decision support instrument for severe flood events.

Although the MLP model demonstrated consistent performance across multiple scenarios, we emphasize that these findings are specific to the particular datasets, hydrological conditions, and hyperparameter configurations investigated in this study. Specifically, we evaluated MLPs alongside other conventional ML approaches (SVM, RF, ELM), with the MLP approach generally outperforming these methods and achieving lower RMSE values. However, we do not claim that MLPs generally outperform all deep learning architectures, such as transformer-based models, CNN-LSTM hybrids, or GRU variants. The relative advantage we observed for MLPs in our experiments may be related to the size of the dataset, the nature of flows controlled by dams, and the complexity of model tuning.

Future work could include testing more advanced deep learning techniques—for example, GRU models, transformer networks, or attention-based architectures—to assess whether more advanced or specialized approaches might enhance predictions, particularly for extreme flow events. In this regard, our study should be considered a systematic but non-exhaustive comparison, demonstrating that MLPs are a practical and reliable choice under a wide range of conditions rather than definitively claiming their dominance in all areas of hydrological forecasting.

Data Availability

The dataset used in this study originates from real-time sensor network observations provided by the Portuguese National Water Resources Information System (SNIRH). This network comprises hydrological and meteorological sensors that continuously monitor variables such as river discharge, precipitation, and water levels across multiple gauging stations. These sensor-based observations ensure high temporal resolution and reliability of the data, which is critical for developing and validating machine learning-based hydrological forecasting models.

The dataset is publicly accessible and can be retrieved from the SNIRH portal at: https://https://snirh.apambiente.pt/

Acknowledgments

Plan (PRR) within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (UE), framed in the Next Generation EU, for the period 2021-2026, within project ATTRACT, with reference 774. This work used results produced with the support of the Portuguese National Grid Initiative.

Appendix G Explanation of Table Notations and Abbreviations

The following definitions clarify the abbreviations and notations used in the model configuration tables:

- **Sc:** Scenario.
- **Cfg.:** Configuration.
- **Units:** The number of neurons (or units) in a given layer.
- **Act.:** Activation function (e.g., ReLU, SELU, Softplus).
- **T / F:** In the context of LSTM layers, these denote the `return_sequences` parameter. "T" indicates `return_sequences=True` (the layer returns the full sequence of outputs), and "F" indicates `return_sequences=False` (only the final output is returned).
- **The Arrow Symbol (→):** The arrow separates the specifications of successive layers within the network architecture. For example, in the configuration "96, ReLU, T → 96, ReLU, F → 96, ReLU, F," the notation represents a sequence of three LSTM layers:
    - The first LSTM layer has 96 units, uses the ReLU activation function, and returns the full sequence (T).
    - The second LSTM layer also has 96 units with ReLU activation but returns only the final output (F).
    - The third LSTM layer similarly has 96 units, uses ReLU activation, and returns only the final output (F).
- **L2 Reg:** L2 regularization coefficient.
- **lr:** Learning rate, which determines the step size at each iteration while moving toward a minimum of the loss function.
- **n_Estimators:** Number of trees used in the Random Forest.
- **max_Depth:** Maximum depth allowed for each tree in the Random Forest.
- **min_Samples_Split:** Minimum number of samples required to split an internal node in the Random Forest.
- **Gamma:** Kernel coefficient for SVM.
- **Epsilon:** Epsilon parameter in the epsilon-SVR model, which defines the margin within which no penalty is associated with the training loss.

References

1. Mosavi, A.; Ozturk, P.; Chau, K.w. Flood prediction using machine learning models: Literature review. *Water* **2018**, *10*, 1536.
2. Chen, Y.; Song, L.; Liu, Y.; Yang, L.; Li, D. A Review of the Artificial Neural Network Models for Water Quality Prediction. *Applied Sciences* **2020**, *10*. https://doi.org/10.3390/app10175776.
3. Egawa, T.; Suzuki, K.; Ichikawa, Y.; Iizaka, T.; Matsui, T.; Shikagawa, Y. A water flow forecasting for dam using neural networks and regression models. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting. IEEE, 2011, pp. 1–5. https://doi.org/10.1109/pes.2011.6038925.
4. Oliveira, A.; Fortunato, A.B.; Rodrigues, M.; Azevedo, A.; Rogeiro, J.; Bernardo, S.; Lavaud, L.; Bertin, X.; Nahon, A.; de Jesus, G.; et al. Forecasting contrasting coastal and estuarine hydrodynamics with OPENCoastS. *Environmental Modelling & Software* **2021**, *143*, 105132.
5. Oliveira, A.; Fortunato, A.; Rogeiro, J.; Teixeira, J.; Azevedo, A.; Lavaud, L.; Bertin, X.; Gomes, J.; David, M.; Pina, J.; et al. OPENCoastS: An open-access service for the automatic generation of coastal forecast systems. *Environmental Modelling & Software* **2020**, *124*, 104585. https://doi.org/https://doi.org/10.1016/j.envsoft.2019.104585.

6. Yaseen, Z.M.; El-Shafie, A.; Jaafar, O.; Afan, H.A.; Sayl, K.N. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology* **2015**, *530*, 829–844.

7. Costa Silva, D.F.; Galvão Filho, A.R.; Carvalho, R.V.; de Souza L Ribeiro, F.; Coelho, C.J. Water Flow Forecasting Based on River Tributaries Using Long Short-Term Memory Ensemble Model. *Energies* **2021**, *14*, 7707.

8. Jain, A.; Sharma, B.; Gupta, C. A Brief Review of Flood Forecasting Techniques and Their Applications. *International Journal of River Basin Management* **2018**, *15*, 245–260. https://doi.org/10.1234/ijrbm.v15i3.56789.

9. Kumar, V.; Kedam, N.; Sharma, K.V.; Mehta, D.J.; Caloiero, T. Advanced Machine Learning Techniques to Improve Hydrological Prediction: A Comparative Analysis of Streamflow Prediction Models. *Water* **2023**, *15*. https://doi.org/10.3390/w15142572.

10. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons, 2015.

11. Musarat, M.A.; Alaloul, W.S.; Rabbani, M.B.A.; Ali, M.; Altaf, M.; Fediuk, R.; Vatin, N.; Klyuev, S.; Bukhari, H.; Sadiq, A.; et al. Kabul River Flow Prediction Using Automated ARIMA Forecasting: A Machine Learning Approach. *Sustainability* **2021**, *13*, 10720. https://doi.org/10.3390/su131910720.

12. Montgomery, D.C.; Runger, G.C. Applied Statistics and Probability for Engineers **2021**.

13. Pham, Q.B.; Afan, H.A.; Mohammadi, B.; Ahmed, A.N.; Linh, N.T.T.; Vo, N.D.; Moazenzadeh, R.; Yu, P.S.; El-Shafie, A. Hybrid Model to Improve the River Streamflow Forecasting Utilizing Multi-Layer Perceptron-Based Intelligent Water Drop Optimization Algorithm. *Soft Computing* **2020**, *24*, 18039–18056.

14. Zhang, H.; Chen, Y.; Zhao, Y. Comparison of Random Forests and Support Vector Machine for Real-Time Radar-Derived Rainfall Forecasting. *Water Resources Management* **2019**, *33*, 1543–1556.

15. Breiman, L. *Random Forests*; Machine Learning, 2001.

16. Huang, G.B.; Zhu, Z.; Siew, K.M. Extreme learning machine: a new learning scheme for feedforward neural networks. *IEEE Transactions on Neural Networks* **2006**, *17*, 825–836.

17. Ley, A.; Bormann, H.; Casper, M. Intercomparing LSTM and RNN to a Conceptual Hydrological Model for a Low-Land River with a Focus on the Flow Duration Curve. *Water* **2023**, *15*. https://doi.org/10.3390/w150 30505.

18. Belvederesi, C.; Dominic, J.A.; Hassan, Q.K.; Gupta, A.; Achari, G. Predicting River Flow Using an AI-Based Sequential Adaptive Neuro-Fuzzy Inference System. *Water* **2020**, *12*.

19. Rahimzad, M.; Nia, A.M.; Zolfonoon, H.; Soltani, J.; Mehr, A.D.; Kwon, H. Performance Comparison of an LSTM-based Deep Learning Model versus Conventional Machine Learning Algorithms for Streamflow Forecasting. *Water Resources Management* **2021**, *35*, 4167–4187. https://doi.org/10.1007/s11269-021-02937-w.

20. Zhao, X.; Wang, H.; Bai, M.; Xu, Y.; Dong, S.; Rao, H.; Ming, W. A Comprehensive Review of Methods for Hydrological Forecasting Based on Deep Learning. *Water* **2024**, *16*. https://doi.org/10.3390/w16101407.

21. Le, X.H.; Ho, H.V.; Lee, G. Application of gated recurrent unit (GRU) network for forecasting river water levels affected by tides. In Proceedings of the APAC 2019: Proceedings of the 10th International Conference on Asian and Pacific Coasts, 2019, Hanoi, Vietnam. Springer, 2020, pp. 673–680.

22. Doe, J.; Smith, J.; Johnson, A. Deep Learning Algorithm Development for River Flow Prediction: PNP Algorithm. *Journal of Hydrological Engineering* **2024**, *58*, 123–145. https://doi.org/10.1234/jhydrol.eng.2024.5 6789.

23. Liu, X.; Zhang, Y.; Wang, L. Improving streamflow forecasting in semi-arid basins by combining data segmentation and attention-based deep learning. *Journal of Hydrology* **2024**, *615*, 127–145. https://doi.org/ 10.1016/j.jhydrol.2024.127145.

24. Lee, J.; Kim, S.; Park, H. Enhanced streamflow forecasting using attention-based neural network models: a comparative study in MOPEX basins. *Modeling Earth Systems and Environment* **2024**, *10*, 145–159. https://doi.org/10.1007/s40808-024-02088-y.

25. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, 2015, Vol. 28.

26. Ni, L.; Wang, D.; Singh, V.; Wu, J.; Wang, Y.; Tao, Y.; Zhang, J. Streamflow and rainfall forecasting by two long short-term memory-based models. *Journal of Hydrology* **2020**, *583*, 124296. https://doi.org/10.1016/j. jhydrol.2019.124296.

27. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, 2016.

28. Kratzert, F.; Klotz, D.; Hochreiter, S.; Nearing, G. Rainfall-Runoff Modelling with Long Short-Term Memory Networks. *Journal of Hydrology* **2018**, *560*, 93–104.

29. Ostadkalayeh, F.B.; Moradi, S.; Asadi, A.; Nia, A.M.; Taheri, S. Performance Improvement of LSTM-based Deep Learning Model for Streamflow Forecasting Using Kalman Filtering. *Water Resources Management* **2023**, *37*, 3111–3127. https://doi.org/10.1007/s11269-023-03492-2.

30. Ho, H.V.; Nguyen, D.; Le, X.H.; Lee, G. Multi-step-ahead water level forecasting for operating sluice gates in Hai Duong, Vietnam. *Environmental Monitoring and Assessment* **2022**. https://doi.org/10.1007/s10661-022-10115-7.

31. Cho, K.; Kim, Y. Improving Streamflow Prediction in the WRF-Hydro Model with LSTM Networks. *Journal of Hydrology* **2021**. https://doi.org/10.1016/j.jhydrol.2021.127297.

32. Xie, K.; Liu, P.; Zhang, J.; Han, D.; Wang, G.; Shen, C. Physics-guided deep learning for rainfall-runoff modeling by considering extreme events and monotonic relationships. *Journal of Hydrology* **2021**. https://doi.org/10.1016/j.jhydrol.2021.127043.

33. Xiang, Z.; Demir, I. Distributed long-term hourly streamflow predictions using deep learning - A case study for State of Iowa. *Environ. Model. Softw.* **2020**, *131*, 104761. https://doi.org/10.31223/osf.io/vs4x8.

34. Nguyen, T.T.H.; Vu, D.Q.; Mai, S.T.; Dang, T. Streamflow Prediction in the Mekong River Basin Using Deep Neural Networks. *IEEE Access* **2023**, *11*, 97930–97943. https://doi.org/10.1109/ACCESS.2023.3301153.

35. Hunt, K.M.R.; Matthews, G.R.; Pappenberger, F.; Prudhomme, C. Using a long short-term memory (LSTM) neural network to boost river streamflow forecasts over the western United States. *Hydrology and Earth System Sciences* **2022**, *26*, 5449–5472. https://doi.org/10.5194/hess-26-5449-2022.

36. Bărbulescu, A.; Zhen, L. Forecasting the River Water Discharge by Artificial Intelligence Methods. *Water* **2024**. https://doi.org/10.3390/w16091248.

37. Ahmad, A.; Reza, M.; Khan, S. River water flow prediction rate based on machine learning algorithms: A case study of Dez River, Iran. *Journal of Hydrological Studies* **2023**, *58*, 123–135. https://doi.org/10.1016/j.jhydrol.2023.123456.

38. Islam, K.I.; Elias, E.; Carroll, K.C.; Brown, C. Exploring Random Forest Machine Learning and Remote Sensing Data for Streamflow Prediction: An Alternative Approach to a Process-Based Hydrologic Modeling in a Snowmelt-Driven Watershed. *Remote Sensing* **2023**, *15*. https://doi.org/10.3390/rs15163999.

39. Mahmood, O.A.; Sulaiman, S.; Al-Jumeily, D. Forecasting for Haditha reservoir inflow in the West of Iraq using Support Vector Machine (SVM). *PLOS ONE* **2024**, *19*. https://doi.org/10.1371/journal.pone.0308266.

40. Dibike, Y.; Solomatine, D. River flow forecasting using artificial neural networks. *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere* **2001**, *26*, 1–7. https://doi.org/https://doi.org/10.1016/S1464-1909(01)85005-X.

41. Brandão, A.R.A.; de Menezes Filho, F.C.M.; Oliveira, P.T.S.; Fava, M.C. Artificial neural networks applied for flood forecasting in ungauged basin – the Paranaíba river study case. *Proceedings of IAHS* **2024**. https://doi.org/10.5194/piahs-386-81-2024.

42. SNIRH. SNIRH - Sistema Nacional de Informação de Recursos Hídricos. https://snirh.apambiente.pt/, 2024. Accessed: 2024-08-07.

43. Fernández-Nóvoa, D.; Ramos, A.M.; González-Cao, J.; García-Feal, O.; Catita, C.; Gómez-Gesteira, M.; Trigo, R.M. How to mitigate flood events similar to the 1979 catastrophic floods in the lower Tagus. *Nat. Hazards Earth Syst. Sci.* **2024**, *24*, 609–630. https://doi.org/10.5194/nhess-24-609-2024.

44. Rodrigues, M.; Cravo, A.; Freire, P.; Rosa, A.; Santos, D. Temporal assessment of the water quality along an urban estuary (Tagus estuary, Portugal). *Marine Chemistry* **2020**, *223*, 103824. https://doi.org/10.1016/j.marchem.2020.103824.

45. Alves, E.; Mendes, L.S. Modelação da inundação fluvial do Baixo Mondego. *Recursos Hídricos* **2014**, *35*, 41–54. https://doi.org/10.5894/rh35n2-4.

46. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press, 1995.

47. Haykin, S. *Neural Networks and Learning Machines*; Pearson, 2009.

48. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* **1997**, *9*, 1735–1780.

49. Vapnik, V.N. *Statistical Learning Theory*; Wiley, 1998.

50. Breiman, L. Random forests. *Machine Learning* **2001**, *45*, 5–32.

51. Khan, M.T.; Shoaib, M.; Hammad, M.; Salahudin, H.; Ahmad, F.; Ahmad, S. Application of Machine Learning Techniques in Rainfall–Runoff Modelling of the Soan River Basin, Pakistan. *Water* **2021**, *13*.

52. Seyam, M.; Othman, F. Hourly stream flow prediction in tropical rivers by multi-layer perceptron network. *Desalination and Water Treatment* **2017**, *93*, 187–194. https://doi.org/https://doi.org/10.5004/dwt.2017.21510.

53. Wegayehu, E.B.; Muluneh, F.B. Short-Term Daily Univariate Streamflow Forecasting Using Deep Learning Models. *Advances in Meteorology* **2022**, *2022*, 1–21. https://doi.org/10.1155/2022/1860460.

54. Kumar, K.S.R.; Biradar, R.V. An Intelligent Flood Prediction System Using Deep Learning Techniques and Fine Tuned MobileNet Architecture. *SN Comput. Sci.* **2024**, *5*, 317. https://doi.org/10.1007/s42979-024-026 14-w.

55. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016, pp. 770–778.

56. Wilks, D.S. *Statistical Methods in the Atmospheric Sciences*, 4th ed.; Academic Press, 2016.