

Article

Forecasting the River Water Discharge by Artificial Intelligence Methods

Alina Bărbulescu ^{1,*}  and Liu Zhen ² ¹ Department of Civil Engineering, Transilvania University of Brasov, 5 Turnului Street, 500152 Brasov, Romania² National Key Laboratory of Deep Oil and Gas, School of Geosciences, China University of Petroleum (East China), Qingdao 266580, China; lincoln110@foxmail.com

* Correspondence: alina.barbulescu@unitbv.ro

Abstract: The management of water resources must be based on accurate models of the river discharge in the context of the water flow alteration due to anthropic influences and climate change. Therefore, this article addresses the challenge of detecting the best model among three artificial intelligence techniques (AI)—backpropagation neural networks (BPNN), long short-term memory (LSTM), and extreme learning machine (ELM)—for the monthly data series discharge of the Buzău River, in Romania. The models were built for three periods: January 1955–September 2006 (S1 series), January 1955–December 1983 (S2 series), and January 1984–December 2010 (S series). In terms of mean absolute error (MAE), the best performances were those of ELM on both Training and Test sets on S2, with $MAE_{Training} = 5.02$ and $MAE_{Test} = 4.01$. With respect to MSE, the best was LSTM on the Training set of S2 (MSE = 60.07) and ELM on the Test set of S2 (MSE = 32.21). Accounting for the R^2 value, the best model was LSTM on S2 ($R^2_{Training} = 99.92\%$, and $R^2_{Test} = 99.97\%$). ELM was the fastest, with 0.6996 s, 0.7449 s, and 0.6467 s, on S, S1, and S2, respectively.

Keywords: river discharge; BPNN; ELM; LSTM

Citation: Bărbulescu, A.; Zhen, L. Forecasting the River Water Discharge by Artificial Intelligence Methods. *Water* **2024**, *16*, 1248. <https://doi.org/10.3390/w16091248>

Academic Editors: Bommana Krishnappan and Guangyuan Kan

Received: 17 March 2024

Revised: 24 April 2024

Accepted: 25 April 2024

Published: 26 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

History indicates that many civilizations appeared and developed along the rivers, the prominent bones of anthropic activity, sustaining food production, transportation, and providing water for drinking and industry [1,2]. On the other hand, floods have often had catastrophic effects on population settlements. Therefore, learning from the past is essential to observe and understand the evolution of the water cycle and river flow dynamic. Setting an early-warning system for taking timely and informed measures to avoid (if possible) or reduce the floods of drought effects on human activities is necessary [3–8]. Research of the rivers' flow was performed in different directions, as follows:

1. Modeling the rivers' discharge. A wide range of techniques have been employed for this purpose:
 - (a) Box–Jenkins methods, such as ARMA, ARIMA, SARIMA [9–13];
 - (b) Statistical approaches [14–16];
 - (c) Artificial intelligence (AI) methods [17–19], such as support vector machines (SVM), artificial neural networks (ANN) [12,20–22], radial basis (RB) neural networks, multi-layer perceptron (MLP) [12], generalized regression neural network, least-square support vector regression [23], long short-term memory (LSTM) [24,25].
 - (d) Hybrid models [10,26], integrating AI and non-linear time series models [27], k-nearest neighbor regression [23], Particle Swarm Optimization–support vector machine (PSO-SVM) [28], Particle Swarm Optimization–long short-term memory PSO-LSTM [29], CEEMDAN-PSO-ELM [30], support vector machine–Particle Swarm Optimization (SVM-PSO) [31], wavelet–autoregressive models [32], wavelet–LSTM [33], etc.;

2. Testing statistical hypotheses [34];
3. Hydraulic modeling [6–8,35];
4. Rainfall-runoff modeling [36,37]
5. Time series decomposition and forecast [38].

For example, statistical analysis [34] can emphasize the series characteristics and the existence of outliers and breakpoints, determine its distribution, and test different hypotheses. It does not effectively provide a river discharge model, which is essential for a reliable forecast. To complement the findings provided by statistical analysis [7,8], the HEC-RAS software was used to model the susceptibility to floods in different basins. HEC-HMS was employed to simulate losses, snowmelt, sub-basin routing, and river flow routing [35].

Rainfall-runoff modeling is a critical aspect of water management, with far-reaching implications for various hydrological processes, particularly those associated with extreme events like flooding. To account for the intricacies of the models, a range of storage parameters are fine-tuned using an extensive dataset of meteorological and hydrological information [37]. The better the model, the better the river discharge forecast.

Various methods can predict future time series behavior, from classical decomposition models—which emphasize trend, seasonality, and random variation to Box–Jenkins methods [8,10,38–40], exponential smoothing [9], and different types of AI [12,20–25] and hybrid models [28–33].

Traditional approaches for modeling and forecasting the river water discharge struggle with large datasets and have limitations in capturing regime changes and non-linearity. Moreover, they are based on restrictive assumptions, like normality or stationarity. By contrast, AI techniques do not impose restrictions on the data; they learn the input series quickly, are less sensitive to the outliers' presence, and can capture abrupt changes in the datasets. These characteristics recommend them as valuable tools to forecast the series' behavior. Comparisons of these two approaches [12,39,41] indicated superior reliability of AI models. Among them, extreme learning machines (ELM) [42,43] proved fast learning features [44]. LSTM [45] can capture the series' long-range dependence, incorporating historical information by automatically learning the data and extracting its future [46]. Backpropagation neural networks (BPNN) proved to be a valuable tool in water-level prediction and were proposed for modeling the river's water discharge [47,48]. Still, there are not enough studies to prove its performance in this research field.

Hydrotechnical structures, such as dams and reservoirs, are crucial for addressing human needs. However, they can significantly alter natural rivers' flow [49–51], sometimes leading to substantial environmental damage and declining biodiversity. Reducing the flood effects on the anthropic settlements was a practical problem that needed to be solved, and it was a study direction for different researchers. Some analyzed the dam's environmental effect, emphasizing the river flow alteration [8,12,52–56]. A deep study of the impact of such structures may contribute to understanding this phenomenon and taking measures to reduce the effects of natural calamities.

The Siriu Dam was built on the upper reach of the Buzău River, one of the most important rivers in Romania, to regulate water flow; reduce the effect of floods on the people living in its catchment; and provide water for drinking, irrigation, and industry. Its importance has not yet been emphasized enough. In Romania, only a few researchers investigated the change in the river flow regimen after the dam inauguration [53–55]. Moreover, the research of rivers' dynamics by AI methods is insignificant, the most utilized tools being statistical methods and hydraulic modeling [5–7,53–55].

The present study is in line with the international literature. Its novelty consists of:

- Providing alternative models for the Buzău River discharge before and after building the dam using AI algorithms;
- Pointing out the modification of the river flow alteration after the dam apparition;
- Exploring the BPNN, LSTM, and ELM capacity for modeling the water river discharge on series with high variability and outliers. These techniques were selected due to their

advantages in modeling time series from various research fields [42–48]. Moreover, we aimed to prove their performances in hydrological modeling (where they were less used compared to other approaches).

- Comparing the single and hybrid AI techniques in hydrological modeling.

2. Study Area and Data Series

Buzău River is one of Romania’s most prominent water bodies. The Buzău River’s catchment (5264 km² and 1043 m average elevation) (Figure 1 (left)) belongs to the Curvature Carpathians, where the climate is temperate—continental. More than four-fifths of the yearly water volume flows upstream of Nehoiu. Catastrophic floods were recorded in the Buzău catchment from 1948, with a maximum discharge of 2100 m³/s in 1975 (one hundred times higher than the average monthly discharge). The Siriu Dam was inaugurated on 1 January 1984. It altered the river’s water discharge and positively impacted the community, reducing the number and intensity of flooding events [55,56].

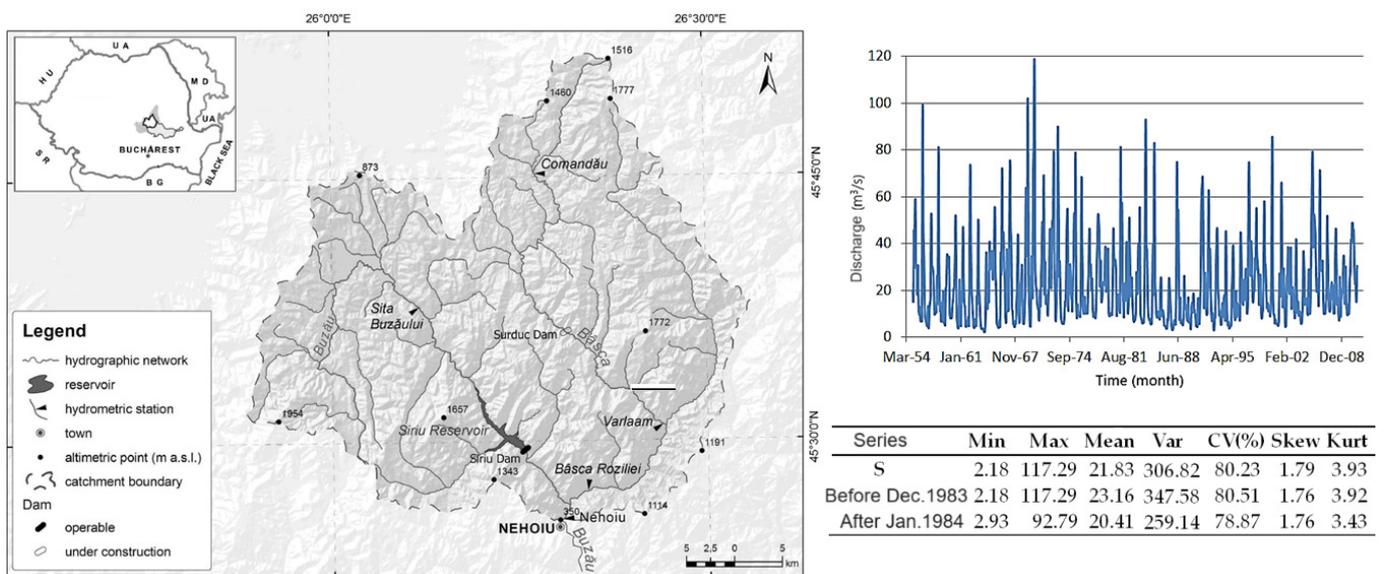


Figure 1. The map of Romania with the position of the Buzău catchment and the catchment’s map containing the position of the Siriu Reservoir and Dam (left), monthly data series (top right), and basic statistics (bottom right). The measurement units for Min, Max, and Mean are m³/s. The variance is expressed in (m³/s)². CV, Skew, and Kurt are dimensionless.

The monthly discharge series from January 1955 to December 2010 (Figure 1 (top right)), denoted in the following by S, are official data from the National Institute of Hydrology and Water Management (INGHA) and contain no missing value. Figure 1 (bottom right) provides the basic statistics of series S and the subseries from January 1995 to December 1983, before building the dam, and from January 1984 to December 2010, after building the dam. Min, Max, and Mean (m³/s) are the minimum, maximum, and mean values. CV, Skew, and Kurt (dimensionless) represent the coefficient of variance, skewness, and kurtosis.

The Mann–Kendall and seasonal Mann–Kendall trend tests did not reject the randomness hypothesis for the monthly series S. Still, they rejected it for the sub-series before December 1983 and after January 1984. The linear trend slopes computed by Sen’s method are 0.0139 for the subseries before December 1983 and 0.0311 for the subseries January 1984–December 2010. So, there is an increasing trend in the river discharge for both subseries but not for the entire series. This result indicates different long-term tendencies of the monthly subseries. The KPSS test did not reject the stationarity hypothesis for all series.

More details on the statistical analyses performed on the monthly Buzău River water discharge may be found in [53–55].

3. Methods

Machine Learning (to which the three algorithms mentioned above belong) is a technique where a computer algorithm can automatically learn from data and make predictions or decisions based on that learning. The typical procedure involves dividing a study series into two parts: the Training set and the Test set. The Training set feeds the algorithm with a dataset, allowing it to learn the relationship between the input features and the target variable. During this stage, the algorithm updates its parameters until it can accurately predict the target variable for new, unseen data. Once the training is complete, the model's performance is evaluated by testing its accuracy on previously unseen data. The Test set objectively measures how well the model performs on new data. The results help to determine if the model is overfitting (performs well on the Training dataset and poorly on the Test set) [57,58].

For our study, we standardized the variables. This study employed the BPNN, LSTM, and ELM algorithms. The Test and Train sets for the models are presented in Table 1.

Table 1. Data set segmentation.

Model	Full Data Range	Training Data Range	Test Data Range
S	January 1955–December 2010	January 1955–December 2005	January 2006–December 2010
S1	January 1955–December 1983	January 1955–December 1983	January 2006–December 2010
S2	January 1984–December 2010	January 1984–December 2005	January 2006–December 2010

Random seeds are essential in computational models involving random search methods like Machine Learning. Their selection can impact the weight initializing and choosing the data sets at different stages of the algorithm. Therefore, setting the same seed is a solution to ensure the reproducibility of the results. To correctly assess the models' performances, they were run with various seeds, which were kept constant in a cycle for all three algorithms.

After obtaining the outputs, the quality of the models was evaluated using the mean absolute error (MAE), mean squared error (MSE), and coefficient of determination (R^2) for both Training and Test sets. The lower (higher) the MAE and MSE, or the closer to 100% (0%) the R^2 , the better (worse) the model is.

To perform the study, we employed Matlab R2023a under Windows 11. The workstation details are as follows: AMD Ryzen 9 5900X 12-Core Processor CPU (3.70 GHz, 12 cores, 24 threads), 64 GB of RAM, NVIDIA GeForce RTX 3090 GPU.

In the following, we present the algorithms involved in computation and the setting used in this study.

3.1. BPNN

A particular type of feed-forward neural network is represented by BPNN [58]. It is an ANN with multiple layers that utilizes the backpropagation algorithm [59] in the learning process. The network is formed by the input, hidden, and output layers. All neurons from the input layer are connected to those in the hidden layer. Weights (w) are assigned to each neuron. The ReLU or sigmoid is used as an activation function. Each input element passes to the next layer, where the new value is computed as a weighted average of the input values in each neuron. Then, it passes to the output layer after applying the activation function [60].

The deviation of the computed values from the recorded ones is measured by the loss function, f , which is MSE. The backpropagation algorithm employed by BPNN updates the weights and minimizes the value of f . The error appearing in the hidden layer is calculated utilizing the chain rule based on the error previously computed for the output layer. The iterations are repeated until the loss function's convergence or the maximum number of iterations previously established is reached. The forecast of the data series is performed only after the training step.

The parameters used to run the BPNN are given in Table 2. In the BPNN model, we utilized for training the Momentum Gradient Descent algorithm, a variant of the standard gradient descent algorithm that accelerates convergence by adding a fraction of the previous update vector to the current update, thereby reducing oscillations and achieving faster convergence.

Table 2. The parameters used to run the BPNN.

Parameter	Value
Number of Input Nodes	1 (Water discharge series)
Number of Hidden Nodes	300
Number of Output Nodes	1 (Water discharge series)
Learning Rate	0.01
Max Iterations	100
Optimization Algorithm	Momentum Gradient Descent
Loss Function	MSE
Shuffle Data Every Epoch	Yes

The learning rate determines the step size during weight updates in optimization. A suitable learning rate is crucial for achieving convergence without overshooting the optimal solution. We used the sigmoid activation function to introduce non-linearity into the network, enabling it to learn and approximate complex relationships within the data.

3.2. LSTM

LSTM [45] is a type of recurrent neural network (RNN) that uses gates to control the flow of information into and out of the network. It also contains a memory cell that keeps information for an extended period. The memory cell works under the control of the Forget, Input, and Output Gates. The Forget Gate selects the information to be discarded from the memory cell, whereas the Input Gate determines what must be added. The Output Gate controls the memory cell's output. This structure allows LSTM to learn long-term dependencies.

The architecture of an LSTM is chain-type with units (Figure 2).

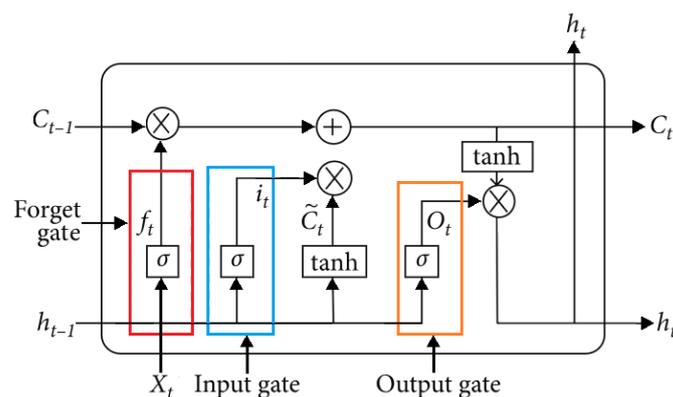


Figure 2. The scheme of a unit of LSTM [61].

The equation that governs the Forget Gate's functioning at a moment j is [45,61–64]:

$$f_j = \sigma(W_f[h_{j-1}, x_j] + b_f), \quad (1)$$

where f_j is the Forget Gate unit at j , W_f is the matrix of the weights of Forget Gate, x_j is the input at j , h_{j-1} is the hidden state at $j - 1$, b_f is the Forget Gate bias at j , and σ is the sigmoid activation function (whose output is 0 or 1, meaning that the information is forgotten or retained).

The Input Gate working at a moment j is performed by:

$$i_j = \sigma(W_i[h_{j-1}, x_j] + b_i), \quad (2)$$

where i_j is the Input Gate's unit at j , W_i is the matrix of the weights of the Input Gate, and b_i is the bias of the Input Gate at j .

In the Input Gate, information passes through a sigmoid function that decides the values to be updated. After that, a hyperbolic tangent function (\tanh) builds the new candidates' vector:

$$\tilde{C}_j = \tanh(W_c[h_{j-1}, x_j] + b_c), \quad (3)$$

where \tilde{C}_j is the candidates' vector, W_c is the matrix of the weights of the candidates' vector, and b_c is the candidates' vector's bias at j .

The equation employed in the update unit to update the information is the following:

$$C_j = f_j * C_{j-1} + i_j * \tilde{C}_j, \quad (4)$$

where $*$ is the element-wise multiplication.

The Output Gate functioning at a moment j is described by:

$$O_j = \sigma(W_o[h_{j-1}, x_j] + b_o), \quad (5)$$

where O_j is the Output Gate unit at j , W_o is the matrix of the Input Gate weights, and b_o is the Output Gate bias at j .

The actual hidden state is calculated by:

$$h_j = O_j * \tanh(C_j). \quad (6)$$

The parameters used to run LSTM are presented in Table 3.

Table 3. The parameters used to run the LSTM.

Parameter	Value
Number of Train Samples	612
Number of Test Samples	131
Max Epochs	100
Initial Learning Rate	0.01
Learning Rate Schedule	Piecewise
Learning Rate Drop Factor	0.1
Learning Rate Drop Period	80% of Max Epochs
Optimization Algorithm	Adam
Shuffle Data Every Epoch	Yes

To train the LSTM network, we utilized the Adam optimizer. Adam is an adaptive learning rate optimization algorithm that combines ideas from momentum and RMSProp methods. It dynamically adjusts the learning rate based on past gradients, leading to faster convergence and improved performance. The batch size used during training, which determines the number of samples processed before updating the model's parameters, was 64. This choice was made to strike a balance between computational efficiency and model convergence. We also conducted tests with 32 and 16, but the best results were obtained with 64.

The dropout regularization technique was applied to prevent overfitting by randomly deactivating a fraction of neurons during training.

3.3. ELM

ELM [42,43,65] is a feed-forward neural network for supervised learning tasks. Its core principles involve the initialization of a neural network and weight learning. ELM has three layers: input, hidden, and output.

The input of the input layer is a vector X of an established dimension (d). The matrix of the weights, W , is randomly initialized in the hidden layer. The hidden layer's output is obtained by passing the product $WX + b$ (b is the bias) through an activation function.

ELM employs the least squares method to learn the weights (O) in the output layer. The O matrix is computed by multiplying the Moore–Penrose pseudo-inverse of the hidden layer output matrix with the class label matrix [66].

The forecast of a vector X' is performed by the formula:

$$y' = g(W'X' + b)O, \quad (7)$$

where W' is the weights matrix from the hidden layer to the output layer.

The results from [41,42] indicate that ELM learns quickly and can act as a universal approximator. ELM is particularly useful when dealing with large datasets because it can train models faster than traditional algorithms. It also has good generalization capacity, so it can make accurate predictions on new, unseen data [66,67].

The parameters utilized to run ELM are shown in Table 4.

Table 4. The parameters used to run the ELM algorithm.

Parameter	Value
Number of Input Nodes	1 (Water discharge series)
Number of Hidden Nodes	300
Number of Output Nodes	1 (Water discharge series)
Activation Function of Hidden Layer	Sigmoid
Input Layer Weight Initialization	Uniform distribution (−1 to 1)
Hidden Layer Weight Initialization	Randomly generated
Input Layer Bias Initialization	0
Hidden Layer Bias Initialization	Random number between 0 and 1
Max Epochs	100
Optimization Algorithm	None
Loss Function	MSE

No specific optimization algorithm was used as the traditional training process of ELM involves randomly initializing the input weights and biases, followed by computing the output weights analytically. Therefore, no iterative optimization process is involved in training ELM.

We used the sigmoid activation function in ELM to introduce non-linearity.

4. Results and Discussion

Figures 3–5 display the recorded and predicted values of the Test set after running BPNN, LSTM, and ELM, respectively, according to the segmentation from Table 1.

Figure 3a shows a high bias between the recorded and computed values in March 2006, February 2006, March 2008, and June 2010 and the discordant behavior of the actual and estimated trend in the neighboring periods. By comparison, in Figure 3b, most values are overestimated, and the estimation errors (difference between the recorded and computed values) are bigger. In Figure 3c, the highest differences between the two series appear in February and September 2007 (56 and 34.5 m³/s, respectively) and February and September 2010 (34 and 29 m³/s, respectively). The visual examination shows that the best fit is provided in Figure 3a. Still, the error from March 2006 (about 72 m³/s) significantly increases the MAE and MSE (that uses the squared errors in computation).

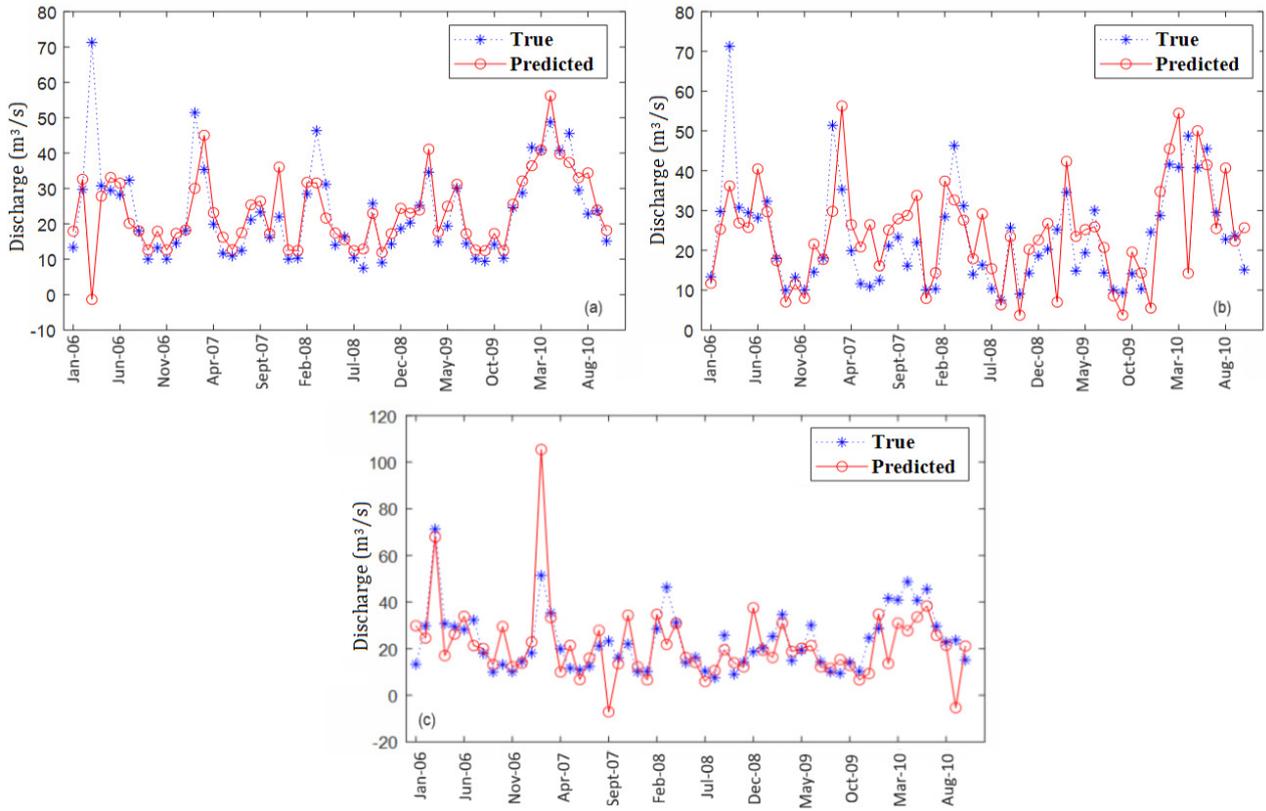


Figure 3. Charts of BPNN's output for the Test set of (a) S, (b) S1, and (c) S2.

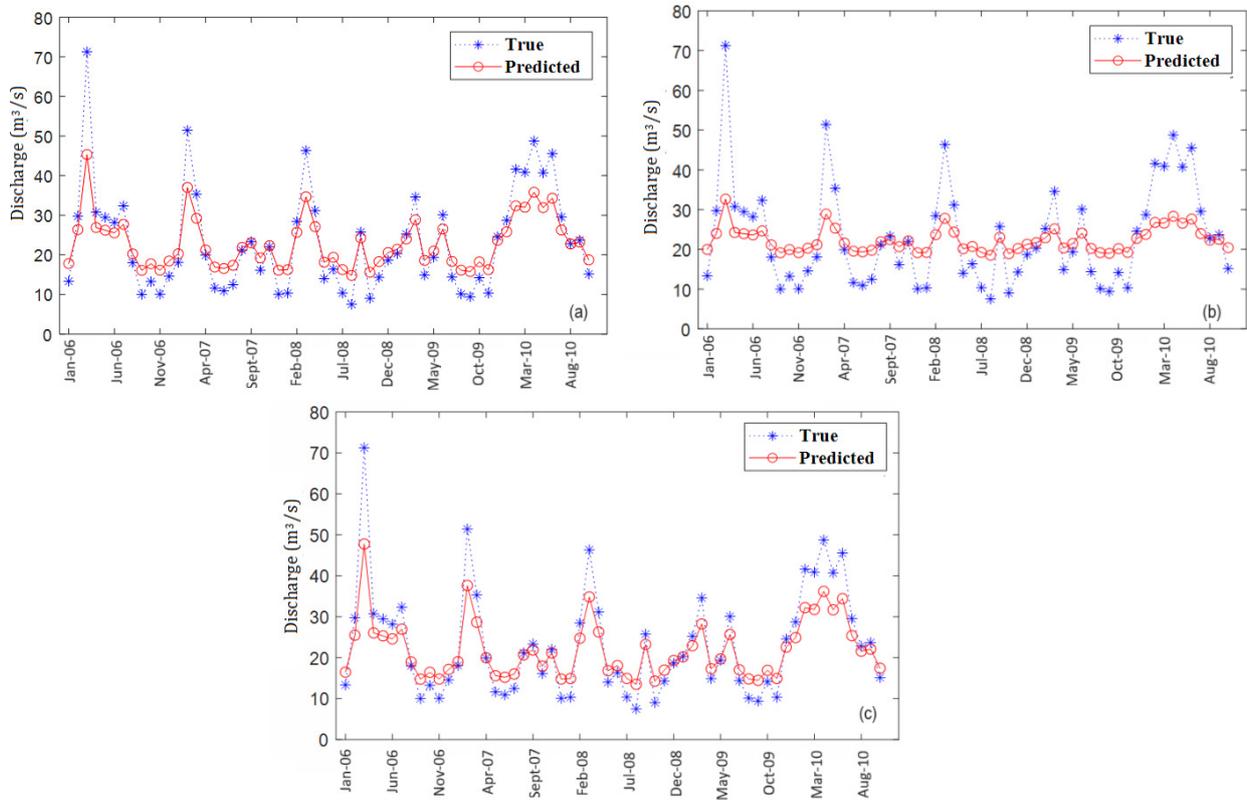


Figure 4. Charts of LSTM's output for the Test set of (a) S, (b) S1, and (c) S2.

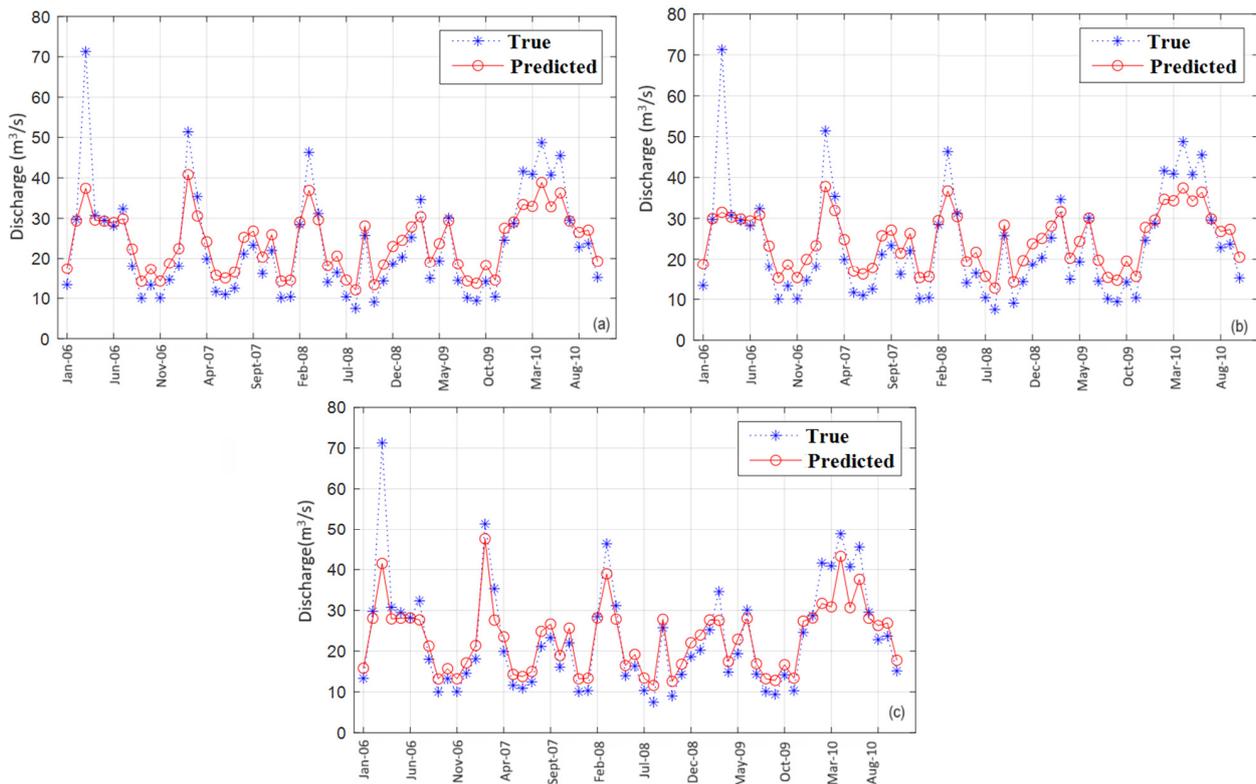


Figure 5. Charts of ELM’s output for the Test set of (a) S, (b) S1, and (c) S2.

Compared to the BPNN, the LSTM models (Figure 4) capture the recorded series pattern better. More precisely, there is no deviation of the forecast trend from the recorded one.

Given that the maximum values of the raw series are underestimated, and some of the minima are overestimated (in Figure 4b), the values of MAE and MSE will not be very small. The forecast series from Figure 4b is the smoothest, so it is expected to have the highest errors compared to Figure 4a,c.

The ELM models (Figure 5) exhibit the highest capacity of capturing the maxima with respect to the competitors. At the same time, the overestimation of the minima is smaller than in the case of BPNN and LSTM, especially for S2. In all cases, the patterns of the original series are entirely captured.

After a visual examination, the output of the ELM algorithm is the best among the three approaches. To confirm this assertion, the goodness-of-fit indicators were computed and are presented in Table 5.

Table 5. The models’ MSE, MAE, and R².

Indicator	Model	Training Set			Test Set		
		S	S1	S2	S	S1	S2
MAE (m ³ /s)	BPNN	6.96	11.00	8.14	5.52	7.94	8.29
	LSTM	6.78	10.50	5.72	4.92	7.64	4.49
	ELM	6.01	6.79	5.02	4.60	5.21	4.01
MSE ((m ³ /s) ²)	BPNN	152.44	326.62	145.38	125.06	116.36	158.55
	LSTM	87.69	213.22	60.07	41.48	98.74	35.65
	ELM	98.12	126.33	78.63	41.29	54.54	32.21
R ² (%)	BPNN	52.89	18.30	0.50	31.07	40.80	42.17
	LSTM	99.39	98.99	99.92	99.83	99.74	99.97
	ELM	83.05	76.14	79.71	88.70	81.84	89.71

The MAE varied between 4.01 (ELM on the S2 Test set) and 11.00 (BPNN on the S1 Training set). All MAEs corresponding to the Training set are higher than those for the Test one, except for S2 on BPNN, indicating that in most cases, the algorithms perform better on the Test sets. In terms of MAE, the best results were given by utilizing ELM: MAE = 5.02 (on the training set) and MAE = 4.01 (on the Test set) on S2. The highest MAEs were 11.00 and 7.94—BPNN on S1 on the Training and Test set, respectively. So, ELM exhibits the best performance. LSTM occupies the second place.

The MSE range is much higher given the existence of the maxima that need to be fitted better, as explained below. On the Training set, MSE varied between 60.07 (S2—LSTM) and 326.62 (S1—BPNN).

On the Test set, MSE belonged to the interval 32.21 (S2—ELM)—158.55 (S2—BPNN). So, in terms of MSE, ELM performed the best on the Test set for S, S1, and S2, and the Training set on S1, whereas on the Training sets for S and S2, the best was LSTM.

R^2 obtained low values in the BPNN model, indicating a reduced concordance between the recorded and forecast series. The values of R^2 were significantly higher in the ELM approach, belonging to the intervals [76.14%, 83.05%] on the Training set and [81.84%, 90.71%] on the Test set. In terms of R^2 , the best performance was achieved by the LSTM algorithm. In this case, R^2 is in the interval [98.99%, 99.97%], pointing out the almost perfect correlation between the actual and predicted values on both Training and Test sets.

Comparing the algorithm capabilities on S, S1, and S2, the best-fitted series is S2. A possible explanation is that S2 is trained, and the forecast is made with values from the same period (after January 1984—the dam inauguration) compared to the model for S, which is trained with values from both periods (before and after building the dam). The fit quality of S1 can be the consequence of the high variability of the river discharge before 1984, attenuated after 1984.

The outputs of algorithms run on S and S1 indicate a better fit in the first case, results expected since S1 was trained on unaltered discharge (before January 1984), but the prediction was made for altered river flow (after January 1984). These findings are concordant with those from [62].

The results reflect the LSTM's ability to capture series nonstationarity and non-linearity, remember information from a long past, and discard irrelevant data. On the other hand, they also indicate the ELM's good generalization capacity [41,42].

Another aspect that was considered is the computation cost (Table 6). Known as a fast learning feed-forward algorithm [41], the ELM achieved the lowest run time. The longest time corresponds to the LSTM on the S series (the longest one), followed by LSTM on S2 and S1 (that have almost the same length). Details on the time complexity of each algorithm can be found in [45,68–70].

Table 6. Time (s) for running the algorithms.

Algorithm	S	S1	S2
BPNN	1.3154	1.2271	1.1639
LSTM	4.3335	3.5738	3.5890
ELM	0.6996	0.7449	0.6467

Comparison of the actual approaches with the existing results on the same data series [62] indicates the following:

- ELM and PSO-ELM perform similarly with respect to all goodness-of-fit indicators. The run time is significantly lower for ELM than that of PSO-ELM (358.07 s for S, 56.43 s for S1, and 50.13 (or S2).
- LSTM was more accurate than CNN-LSTM in terms of R^2 and MSE for S and S2. The run time for LSTM was 2.35 (1.63) times lower than that of CNN-LSTM for S (S2).
- All algorithms perform better than multi-layer perceptron (MLP) and the Box–Jenkins (ARIMA) models for S, S1, and S2.

These results confirm the suitability of LSTM for modeling hydrological series [71,72] and point out the ELM [73,74] as a possible competitor in solving such problems.

5. Conclusions

The research presented here examined the performances of three AI models for forecasting the Buzău River water discharge for 572 months and subperiods before and after building the Siriu dam and pointed out the river flow alteration after 1984 without using statistical tools. It also compared the goodness of fit of BPNN, LSTM, and ELM with the same series' PSO-ELM, CNN-LSTM, MLP, and ARIMA models.

This research makes a unique contribution to the field by testing the performances of the first three mentioned algorithms on the river flow series. This novel approach aims to enhance their abilities to learn patterns and forecast datasets with high variability, thereby introducing a new dimension to the field of water resource management.

Various performances of different algorithms were expected, given their architectures and functioning. ELM performed best in terms of MAE (between 4.01 (S2—Test) and 6.79 (S1 Training)) and MSE (between 32.21 (S2—Test) and 98.79 (S1—Training)). With respect to R^2 , ELM placed second, with values between 79.71% (S—Training) and 89.71% (S2—Test), after LSTM, with R^2 in the interval 98.99% (S1—Training) and 99.97% (S2—Test). ELM was also the fastest, with a runtime under 0.75 s in all cases, indicating that it quickly learns of the series patterns and accurately applies what it knows. LSTM confirmed its capacity to preserve the long-term essential information and its good reproduction capacity of the non-linearities in the data series. By comparison with the ELM, it was at least six times slower.

Given that ELM is the best based on MAE and MSE, has a low computational cost, and has a high R^2 , it is recommended to achieve the study objective.

The second novelty is comparing single and hybrid models on the same data series. Based on R^2 and MSE, LSTM is better than the hybrid competitors and has a significantly lower computational cost. So, it is recommended for modeling the study series. The worst performances among all techniques were those of MLP and ARIMA. These findings reconfirm the capacity of AI methods in forecasting modeling hydrological series.

One of the things learned from these models is that not all AI algorithms have the ability to predict with very high accuracy any changes in river flow due to external effluences such as a dam. Moreover, AI models should be incorporated into physical models to be of value beyond the traditional models when changes occur in the watershed.

The best results were obtained on S2 (the shortest series), for which both Training and Test sets belong to the same period (after January 1984). The worst models' performances were achieved on the S1, for which the Training was performed on a dataset before December 1983 and the Test on a dataset after January 1984. This proves that the pattern learned in the first set is not the same as that of the second one, so the alteration of the river flow was significant after building the dam.

The next stage of the work is to apply the same techniques to daily data series. Given the computational complexity, we expect some algorithms to have high run times and require more powerful computational tools. Moreover, data preprocessing will be necessary, given the episodes of high flows that were not captured by the monthly series.

Author Contributions: Conceptualization, A.B. and L.Z.; methodology, A.B. and L.Z.; software, L.Z.; validation, A.B.; formal analysis, A.B. and L.Z.; investigation A.B. and L.Z.; resources, A.B.; data curation, A.B. and L.Z.; writing—original draft preparation, A.B.; writing—review and editing, A.B.; visualization, L.Z.; supervision, A.B.; project administration, A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: The research received no funding.

Data Availability Statement: Data will be available on request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van De Wiel, M.J.; Coulthard, T.J.; Macklin, M.G.; Lewin, J. Modelling the response of river systems to environmental change: Progress, problems and prospects for palaeo-environmental reconstructions. *Earth Sci. Rev.* **2011**, *104*, 167–185. [[CrossRef](#)]
2. Bărbulescu, A.; Barbeș, L.; Dumitriu, C.Ș. Statistical Assessment of the Water Quality Using Water Quality Indicators—Case Study from India. In *Water Safety, Security and Sustainability. Advanced Sciences and Technologies for Security Applications*; Vaseashta, A., Maftai, C., Eds.; Springer: Cham, Switzerland, 2021; pp. 599–613.
3. Bărbulescu, A.; Maftai, C.E. Evaluating the Probable Maximum Precipitation. Case study from the Dobrogea region, Romania. *Rom. Rep. Phys.* **2023**, *75*, 704. [[CrossRef](#)]
4. Bărbulescu, A.; Dumitriu, C.S.; Maftai, C. On the Probable Maximum Precipitation Method. *Rom. J. Phys.* **2022**, *67*, 801.
5. Crăciun, A.; Costache, R.; Bărbulescu, A.; Chandra Pal, S.; Costache, I.; Dumitriu, C.S. Modern techniques for flood susceptibility estimation across the Deltaic Region (Danube Delta) from the Black Sea's Romanian Sector. *J. Marine Sci. Eng.* **2022**, *10*, 1149. [[CrossRef](#)]
6. Popescu, C.; Bărbulescu, A. On the Flash Flood Susceptibility and Accessibility in the Vărbilău Catchment (Romania). *Rom. J. Phys.* **2022**, *67*, 811.
7. Popescu, C.; Bărbulescu, A.; Dumitriu, C.S. Modeling Road Accessibility in a Flood-Prone Area in Romania. *Eng. Proc.* **2023**, *39*, 22. [[CrossRef](#)]
8. Ahmadpour, A.; Mirhashemi, S.H.; Haghghatjou, P.; Foroughi, F. Comparison of the monthly streamflow forecasting in Maroon dam using HEC-HMS and SARIMA models. *Sustain. Water Resour. Manag.* **2022**, *8*, 158. [[CrossRef](#)]
9. Ghimire, B.N. Application of ARIMA Model for River Discharges Analysis. *J. Nepal Phys. Soc.* **2017**, *4*, 27–32. [[CrossRef](#)]
10. Phan, T.-T.-H.; Nguyen, X.H. Combining statistical machine learning models with ARIMA for water level forecasting: The case of the Red river. *Adv. Water Resour.* **2020**, *142*, 103656. [[CrossRef](#)]
11. Subha, J.; Saudia, S. Robust Flood Prediction Approaches Using Exponential Smoothing and ARIMA Models. In *Artificial Intelligence and Sustainable Computing*; Pandit, M., Gaur, M.K., Kumar, S., Eds.; Springer: Singapore, 2023; pp. 457–470.
12. Valipour, M.; Banihabib, M.E.; Behbahani, S.M.R. Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *J. Hydrol.* **2013**, *476*, 433–441. [[CrossRef](#)]
13. Zhang, X.; Wu, X.; Zhu, G.; Lu, X.; Wang, K. A seasonal ARIMA model based on the gravitational search algorithm (GSA) for runoff prediction. *Water Supply* **2022**, *22*, 6959–6977. [[CrossRef](#)]
14. Yürekli, K.; Kurunc, A.; Ozturk, F. Application of Linear Stochastic Models to Monthly Flow Data of Kelkit Stream. *Ecol. Model.* **2005**, *183*, 67–75. [[CrossRef](#)]
15. Uca; Toriman, E.; Jaafar, O.; Maru, R.; Arfan, A.; Ahmar, A.S. Daily Suspended Sediment Discharge Prediction Using Multiple Linear Regression and Artificial Neural Network. *J. Phys. Conf. Ser.* **2018**, *954*, 012030. [[CrossRef](#)]
16. Chaibandit, K.; Konyai, S. Using Statistics in Hydrology for Analyzing the Discharge of Yom River. *APCBEE Procedia* **2012**, *1*, 356–362. [[CrossRef](#)]
17. Dumitriu, C.S.; Bărbulescu, A. Artificial intelligence models for the mass loss of copper-based alloys under the cavitation. *Materials* **2022**, *15*, 6695. [[CrossRef](#)] [[PubMed](#)]
18. Bărbulescu, A.; Dumitriu, C.S. Modeling the Voltage Produced by Ultrasound in Seawater by Stochastic and Artificial Intelligence Methods. *Sensors* **2022**, *22*, 1089. [[CrossRef](#)]
19. Dumitriu, C.Ș.; Dragomir, F.-L. Modeling the Signals Collected in Cavitation Field by Stochastic and Artificial Intelligence Methods. In Proceedings of the 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 1–3 July 2021; pp. 1–4.
20. Alquraish, M.M.; Khadr, M. Remote-Sensing-Based Streamflow Forecasting Using Artificial Neural Network and Support Vector Machine Models. *Remote Sens.* **2021**, *13*, 4147. [[CrossRef](#)]
21. Kisi, Ö.; Cobaner, M. Modeling River Stage-Discharge Relationships Using Different Neural Network Computing Techniques. *Clean* **2009**, *37*, 160–169. [[CrossRef](#)]
22. Tanty, R.; Desmukh, T.S. Application of Artificial Neural Network in Hydrology—A Review. *Int. J. Eng. Resear. Technol.* **2015**, *4*, 184–188.
23. Modaresi, F.; Araghinejad, S.; Ebrahimi, K. A comparative assessment of artificial neural network, least-square support vector regression, and K-nearest neighbor regression for monthly streamflow forecasting in linear and nonlinear conditions. *Water Resour. Manag.* **2018**, *32*, 243–258. [[CrossRef](#)]
24. Kratzert, F.; Klotz, D.; Brenner, C.; Schulz, K.; Herrnegger, M. Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrol. Earth Sys. Sci.* **2018**, *22*, 6005–6022. [[CrossRef](#)]
25. Ni, L.; Wang, D.; Singh, V.P.; Wu, J.; Wang, Y.; Tao, Y.; Zhang, J. Streamflow and rainfall forecasting by two long short-term memory-based models. *J. Hydrol.* **2019**, *583*, 124296. [[CrossRef](#)]
26. Xu, H.; Song, S.; Li, J.; Guo, T. Hybrid model for daily runoff interval predictions based on Bayesian inference. *Hydrol. Sci. J.* **2022**, *68*, 62–75. [[CrossRef](#)]
27. Fathian, F.; Mehdizadeh, S.; Sales, A.K.; Safari, M.J.S. Hybrid models to improve the monthly river flow prediction: Integrating artificial intelligence and non-linear time series models. *J. Hydrol.* **2019**, *575*, 1200–1213. [[CrossRef](#)]
28. Samantaray, S.; Sahoo, A.; Agnihotri, A. Prediction of Flood Discharge Using Hybrid PSO-SVM Algorithm in Barak River Basin. *MethodsX* **2023**, *10*, 102060. [[CrossRef](#)]

29. Ruma, J.F.; Adnan, M.S.G.; Dewan, A.; Rahman, M.R. Particle swarm optimization based LSTM networks for water level forecasting: A case study on Bangladesh river network. *Result Eng.* **2023**, *17*, 100951. [CrossRef]
30. Zhang, X.Q.; Zhao, D.; Wang, T.; Wu, X.L.; Duan, B.S. A novel rainfall prediction model based on CEEMDAN-PSO-ELM coupled model. *Water Supply* **2023**, *22*, 4531–4543. [CrossRef]
31. Zaini, N.; Malek, M.A.; Yusoff, M.; Mardi, N.H.; Norhisham, S. Daily River Flow Forecasting with Hybrid Support Vector Machine—Particle Swarm Optimization. *IOP Conf. Ser. Earth Environ. Sci.* **2018**, *140*, 012035. [CrossRef]
32. Tantanee, S.; Patamatammakul, S.; Oki, T.; Sriboonlue, V.; Prempre, T. Coupled Wavelet-Autoregressive Model for Annual Rainfall Prediction. *J. Environ. Hydrol.* **2005**, *13*, 1–8.
33. Liang, Z.; Liu, Y.; Hu, H.; Li, H.; Ma, Y.; Khan, M.Y.A. Combined Wavelet Transform with Long Short-Term Memory Neural Network for Water Table Depth Prediction in Baoding City, North China Plain. *Front. Environ. Sci.* **2021**, *9*, 7804. [CrossRef]
34. Bărbulescu, A.; Dumitriu, C.S. Assessing the water quality by statistical methods. *Water* **2021**, *13*, 1026. [CrossRef]
35. Pang, Y.-H.; Wang, H.-B.; Zhao, J.-J.; Shang, D.-Y. Analysis and Prediction of Hydraulic Support Load Based on Time Series Data Modeling. *Geofluids* **2020**, *2020*, 8851475. [CrossRef]
36. Kapoor, A.; Pathiraja, S.; Marshall, L.; Chandra, R. DeepGR4]: A deep learning hybridization approach for conceptual rainfall-runoff modelling. *Environ. Modell. Softw.* **2023**, *169*, 105831. [CrossRef]
37. Kratzert, F.; Klotz, D.; Herrnegger, M.; Sampson, A.K.; Hochreiter, S.; Nearing, G.S. Towards Improved Predictions in Ungauged Basins: LSTM Networks for Rainfall-Runoff Modeling. *Water Resour. Res.* **2019**, *55*, 11344–11354. [CrossRef]
38. Alonso Brito, G.R.; Rivero Villaverde, A.; Lau Quan, A.; Ruíz Pérez, M.E. Comparison between SARIMA and Holt–Winters models for forecasting monthly streamflow in the western region of Cuba. *SN Appl. Sci.* **2021**, *3*, 671. [CrossRef]
39. Abraham, R.J.; See, L. Comparing Neural Network and Autoregressive Moving Average Techniques for the Provision of Continuous River Flow Forecasts in Two Contrasting Catchments. *Hydrol. Process.* **2000**, *14*, 2157–2172. [CrossRef]
40. Khan, F.; Pilz, J. Modelling and sensitivity analysis of river flow in the Upper Indus Basin, Pakistan. *Int. J. Water* **2018**, *12*, 1–21. [CrossRef]
41. Birikundavi, S.; Labib, R.; Trung, H.T.; Rousselle, J. Performance of Neural Networks in Daily Streamflow Forecasting. *J. Hydrol. Eng.* **2002**, *7*, 392. [CrossRef]
42. Huang, G.-B. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
43. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
44. Ahuja, B.; Vishwakarma, V.P. Deterministic Multi-kernel based extreme learning machine for pattern classification. *Expert Syst. Appl.* **2021**, *183*, 115308. [CrossRef]
45. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
46. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches. *Energies* **2018**, *11*, 1636. [CrossRef]
47. Ghose, D.K. Measuring Discharge Using Back-Propagation Neural Network: A Case Study on Brahmani River Basin. In *Intelligent Engineering Informatics*; Bhateja, V., Coello, C.A.C., Satapathy, S.C., Pattnaik, P.K., Eds.; Springer: Singapore, 2018; pp. 591–598.
48. Khan, M.Y.A.; Hasan, F.; Panwar, S.; Chakrapani, G.J. Neural network model for discharge and water-level prediction for Ramganga River catchment of Ganga Basin, India. *Hydrol. Sci. J.* **2016**, *61*, 2084–2095. [CrossRef]
49. Magilligan, F.J.; Nislow, K.H. Changes in hydrologic regime by dams. *Geomorphology* **2005**, *71*, 61–78. [CrossRef]
50. Nislow, K.H.; Magilligan, F.; Fassnacht, H.; Becgtel, D.; Ruesink, A. Effects of Dam Impoundment on the Flood Regime of Natural Floodplain Communities in the Upper Connecticut River. *JAWRA J. Am. Water Res. Assoc.* **2002**, *38*, 1533–1548. [CrossRef]
51. Richter, B.D.; Baumgartner, J.V.; Powell, J.; Braun, D.P. A method for assessing hydrologic alteration within ecosystems. *Conserv. Biol.* **1996**, *10*, 1163–1174. [CrossRef]
52. Bisoyi, N.; Gupta, N.; Padhy, N.P.; Chakrapani, G.J. Prediction of daily sediment discharge using a back propagation neural network training algorithm: A case study of the Narmada River, India. *Int. J. Sedim. Resear.* **2019**, *34*, 125–135. [CrossRef]
53. Minea, G.; Bărbulescu, A. Statistical assessing of hydrological alteration of Buzău River induced by Siriu dam (Romania). *Forum Geogr.* **2014**, *13*, 50–58. [CrossRef]
54. Mocanu-Vargancsik, C.; Tudor, G. On the linear trends of a water discharge data under temporal variation. Case study: The upper sector of the Buzău river (Romania). *Forum Geogr.* **2020**, *19*, 37–44. [CrossRef]
55. Chendeş, V. *Water Resources in Curvature Subcarpathians. Geospatial Assessments*; Editura Academiei Române: Bucureşti, Romania, 2011. (In Romanian with English Abstract)
56. The Arrangement of the Buzău River. Available online: https://www.hidroconstructia.com/dyn/2pub/proiecte_det.php?id=110&pg=1 (accessed on 17 October 2023). (In Romanian)
57. Difference between Training Data and Testing Data. Available online: <https://edupepper.com/difference-between-training-data-and-testing-data/> (accessed on 17 April 2024).
58. Fausett, L. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*; Prentice-Hall Inc.: Upper Saddle River, NJ, USA, 1994.
59. Rumelhart, D.; Hinton, G.; Williams, R. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]

60. Wang, W.; Du, Y.; Chau, K.; Chen, H.; Liu, C.; Ma, Q. A Comparison of BPNN, GMDH, and ARIMA for Monthly Rainfall Forecasting Based on Wavelet Packet Decomposition. *Water* **2021**, *13*, 2871. [CrossRef]
61. Saxena, S. What is LSTM? *Introduction to Long Short-Term Memory*. Available online: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/> (accessed on 17 March 2024).
62. Zhen, L.; Bărbulescu, A. Comparative Analysis of Convolutional Neural Network-Long Short-Term Memory, Sparrow Search Algorithm-Backpropagation Neural Network, and Particle Swarm Optimization-Extreme Learning Machine for the Water Discharge of the Buzău River, Romania. *Water* **2024**, *16*, 289. [CrossRef]
63. Deep Learning | Introduction to Long Short Term Memory. Available online: <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/> (accessed on 17 March 2024).
64. Understanding LSTM Networks. Available online: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed on 17 March 2024).
65. Extreme Learning Machine. Available online: <https://www.geeksforgeeks.org/extreme-learning-machine/> (accessed on 17 March 2024).
66. Zhu, H.; Tsang, E.C.C.; Zhu, J. Training an extreme learning machine by localized generalization error model. *Soft Comput.* **2018**, *22*, 3477–3485. [CrossRef]
67. Zhu, B.; Feng, Y.; Gong, D.; Jiang, S.; Zhao, L.; Cui, L. Hybrid particle swarm optimization with extreme learning machine for daily reference evapotranspiration prediction from limited climatic data. *Comput. Electron. Agr.* **2020**, *173*, 105430. [CrossRef]
68. Tsironi, E.; Barros, P.; Weber, C.; Wermter, S. An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition. *Neurocomputing* **2017**, *268*, 76–86. [CrossRef]
69. Zhang, R.; Pan, Z.; Yin, Y.; Cai, Z. A Model of Network Security Situation Assessment Based on BPNN Optimized by SAA-SSA. *Int. J. Digital Crime Forens.* **2022**, *14*, 1–18. [CrossRef]
70. Karlsson, V.; Rosvall, E. Extreme Kernel Machine. Available online: <https://www.diva-portal.org/smash/get/diva2:1130092/FULLTEXT01.pdf> (accessed on 6 January 2024).
71. Ouma, Y.O.; Cheruyot, R.; Wachera, A.N. Rainfall and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: Case study of Nzoia hydrologic basin. *Complex Intell. Syst.* **2022**, *2022*, 213–236. [CrossRef]
72. Dai, Z.; Zhang, M.; Nedjah, N.; Xu, D.; Ye, F. A Hydrological Data Prediction Model Based on LSTM with Attention Mechanism. *Water* **2023**, *15*, 670. [CrossRef]
73. Liu, T.; Ding, Y.; Cai, X.; Zhu, Y.; Zhang, X. Extreme learning machine based on particle swarm optimization for estimation of reference evapotranspiration. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 4567–4572.
74. Anupam, S.; Pani, P. Flood forecasting using a hybrid extreme learning machine-particle swarm optimization algorithm (ELM-PSO) model. *Model. Earth Syst. Environ.* **2020**, *6*, 341–347. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.