water

MDPI

Article

Comparative Analysis of Convolutional Neural Network-Long Short-Term Memory, Sparrow Search Algorithm-Backpropagation Neural Network, and Particle Swarm Optimization-Extreme Learning Machine Models for the Water Discharge of the Buzău River, Romania

Liu Zhen [1,2,3] and Alina Bărbulescu [1,*]

[1] Department of Civil Engineering, Transilvania University of Brasov, 5, Turnului Street, 500152 Brasov, Romania; lincoln110@foxmail.com
[2] National Key Laboratory of Deep Oil and Gas, China University of Petroleum (East China), Qingdao 266580, China
[3] School of Geosciences, China University of Petroleum (East China), Qingdao 266580, China
* Correspondence: alina.barbulescu@unitbv.ro

Abstract: Modeling and forecasting the river flow is essential for the management of water resources. In this study, we conduct a comprehensive comparative analysis of different models built for the monthly water discharge of the Buzău River (Romania), measured in the upper part of the river's basin from January 1955 to December 2010. They employ convolutional neural networks (CNNs) coupled with long short-term memory (LSTM) networks, named CNN-LSTM, sparrow search algorithm with backpropagation neural networks (SSA-BP), and particle swarm optimization with extreme learning machines (PSO-ELM). These models are evaluated based on various criteria, including computational efficiency, predictive accuracy, and adaptability to different training sets. The models obtained applying CNN-LSTM stand out as top performers, demonstrating a superior computational efficiency and a high predictive accuracy, especially when built with the training set containing the data series from January 1984 (putting the Siriu Dam in operation) to September 2006 (Model type S2). This research provides valuable guidance for selecting and assessing river flow prediction models, offering practical insights for the scientific community and real-world applications. The findings suggest that Model type S2 is the preferred choice for the discharge forecast predictions due to its high computational speed and accuracy. Model type S (considering the training set recorded from January 1955 to September 2006) is recommended as a secondary option. Model type S1 (with the training period January 1955–December 1983) is suitable when the other models are unavailable. This study advances the field of water discharge prediction by presenting a precise comparative analysis of these models and their respective strengths

Keywords: flow prediction; CNN-LSTM; SSA-BP; PSO-ELM

check for updates

Citation: 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 Models for the Water Discharge of the Buzău River, Romania. *Water* **2024**, *16*, 289. https://doi.org/10.3390/w16020289

Academic Editor: Giuseppe Pezzinga

Received: 2 December 2023
Revised: 9 January 2024
Accepted: 11 January 2024
Published: 15 January 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

From ancient times, rivers are places along which civilizations developed, providing water for consumption, agriculture, transportation, and other activities [1]. Understanding their dynamics is necessary, given their complex role in communities' existence and as a main background for water management policies [2–7]. Studying river discharge in correlation with other environmental variables will lead to a better understanding of climate change [8–11].

Different techniques have been used for modeling the rivers' discharge. Rahayu et al. [12] modeled the Amprong River discharge using an autoregressive integrated moving average (ARIMA) approach. Ghimire [13] applied the same technique in two case studies from

the USA, whereas Valipour [14] proposed two alternative models, ARIMA and seasonal ARIMA (SARIMA), for long-term runoff analysis. Yürekli et al. [15] used ARIMA to simulate the monthly discharge of Kelkit Stream.

Conventional flood prediction approaches, often reliant on empirical hydrological and meteorological models, struggle with large-scale and complex data sets. Despite ARIMA models being simple, easy to implement, and flexible, capturing the series components (trend, seasonality, and cycles), they cannot handle nonlinearities, regime changes, or shocks. Moreover, specific hypotheses must be fulfilled using data series and residuals. Since the model's quality may be affected by outliers or missing existence, data preprocessing is necessary before modeling [16]. Therefore, other approaches have been proposed to address these drawbacks. Some of them are artificial intelligence (AI), or machine learning (ML), models that do not consider mathematical relationships, utilizing only sets of input parameters, in contrast with the physical models that utilize mathematical tools to predict hydrological phenomena [17].

AI technology has become a research hotspot in engineering and science fields in recent years due to its significant capabilities in handling big data, pattern recognition, automated decision making, and predictive modeling, as well as enhancing efficiency and accuracy. AI models predict natural disasters, aiding in early preparation and the mitigation of their impacts. Therefore, ML techniques have attracted the attention of scientists working in water resources. For example, Abrahart and See [18] compared the forecasting power of ANN and ARMA models of river flow data for two catchments. Birikundavyi et al. [19] compared the performances of artificial neural networks (ANNs) and autoregressive moving average (ARMA) techniques in predicting the daily streamflow and showed better results obtained by the first approach. Hong and Hong [20] employed ANN to forecast the flooding produced by a river in Malaysia. Kisi and Çobaner [21] employed multi-layer perceptron (MLP) and radial basis (RB) neural networks to model flow series recorded at three stations on Kizilirmak River (Turkey) and study river stage–discharge relationships using different neural network computing techniques. A review of the ANN applications in hydrology can be found in [22]. Valipour et al. [23] compared the forecast of the Dez Dam Reservoir monthly inflow obtained using ARIMA, ARMA, and autoregressive artificial neural networks. They found that the best forecasting model was the dynamic artificial neural network with a sigmoid activation function. Uca et al. [24] compared multiple linear regression (MLRg) and ANN in the discharge prediction of the Jenderam, showing that the first approach had the best performance.

Combined approaches have also been proposed to benefit from the capabilities of various techniques. Li and Yang [25] employed a Bayesian optimized ML seasonally adjusted to model the suspended sediment load. Hayder et al. [26] proposed the use of particle swarm-optimized cascade-forward neural networks on a case study from Malaysia. Xiang et al. [27] introduced an adaptive intelligent dynamic water planning (AIDWRP) model to optimize environmental planning.

During the last period, models combining convolutional neural networks with long short-term memory (CNN-LSTM), the sparrow search algorithm with backpropagation neural networks (SSA-BP), and particle swarm optimization with extreme learning machines (PSO-ELM) provided very good results in various fields.

CNN-LSTM, an innovative deep learning architecture, has achieved breakthrough results in fields such as image and speech recognition and natural language processing. It combines the spatial feature extraction capabilities of CNNs with the sequential data processing strength of long short-term memory (LSTM), effectively handling complex series data. For instance, Essien et al. [28] utilized the CNN-LSTM framework to predict urban traffic flow, achieving higher accuracy and efficiency than traditional methods. Zhang and Li [29] developed a CNN-LSTM model to enhance the accuracy of air quality forecasting. This model outperformed SARIMA.

The SSA-BP method merges the global search capability of the sparrow search algorithm (SSA) with the powerful learning mechanism of backpropagation neural networks

(BP). This approach has shown exceptional performance in power systems, financial market analysis, and bioinformatics. For example, Yan et al. [30] successfully employed an SSA-optimized BP neural network for classifying potential water sources for coal mines. Xin et al. [31] introduced a BP neural network model optimized with the sparrow search algorithm (SSA) to identify pipeline deformation.

PSO-ELM combines the efficient global search capability of particle swarm optimization (PSO) with the rapid learning features of extreme learning machines (ELMs). This combination has demonstrated strong potential in complex problems like predicting the performance of building materials [32].

Zhang et al. [33] developed a CEEMDAN-PSO-ELM approach and applied it to monthly precipitation forecasting. A comparative analysis with LSTM, ELM, and PSO-ELM highlighted its significant benefits in hydrological simulation and prediction.

The hybrid algorithms have shown significant advantages over traditional methods in river flow forecasting. Still, Kratzert et al. [34,35] demonstrated the effectiveness of LSTM in flood forecasting, highlighting its superiority in prediction accuracy and laying the groundwork for applying more complex hybrid methods like CNN-LSTM in flood prediction.

Our search of the scientific literature yielded insignificant results on modeling water discharges based on CNN-LSTM, SSA-BP, or PSO-ELM, despite the proven performances of these approaches.

Our research aims to answer whether the Buzău River discharge is altered after putting the Siriu Dam, one of Romania's most important accumulation lakes in the country, into operation. Two articles [36,37] attempted to answer this question by testing different statistical hypotheses and using indicators of hydrologic alterations (IHA). Two models (regression and generalized regression neural network) [37,38] for the daily river discharge have also been proposed, but neither of them was satisfactory from an accuracy viewpoint. Given the importance of predicting the river discharge (based on correct models) for the Romanian Risk Management Plan, this paper provides three alternative models for the monthly discharge of the Buzău River. The significance of this approach consists of the following.

(1) It provides reliable models for the monthly discharge of the Buzău River for the first time.

(2) It emphasizes that building the Siriu Dam impacted the river flow, confirming the findings of the statistics from [37].

(3) It analyzes and compares the effectiveness of CNN-LSTM, SSA-BP, and PSO-ELM in the river's water discharge forecasting field. From this point of view, these approaches are new in the hydrological series modeling.

Moreover, the potential and advantages of these advanced algorithms in water resources modeling and forecasting are demonstrated, and new perspectives and directions for future research and practice are provided.

2. Materials and Methods

2.1. Study Area and Data Series

Hydrotechnical arrangements, like dams and water accumulations, are built to solve anthropic needs and avoid catastrophic events, diminishing flooding frequencies and intensity. The Siriu Dam, on the Buzău River in Romania, was constructed for such reasons.

The Buzău River is one of the most important rivers in Romania from the viewpoint of the population served for drinking water, agricultural, and industrial uses. The principal floods on the Buzău River were recorded in 1948, 1969, 1971, 1975 (with a peak flow of 2100 m^3/s), 1980, 1984, 1991, and 2005 in May–July. The floods were very frequent, with high intensities, upstream of Nehoiu city, before the Siriu Dam, the second largest embankment dam in Romania, was installed.

The Buzău River's catchment (Figure 1) is located in a temperate–continental climate and covers a surface of 5264 km^2. The river basin's mean elevation is 1043 m. In the natural regimen, the river's flow is between 0.76 m^3/s and 5000 m^3/s.

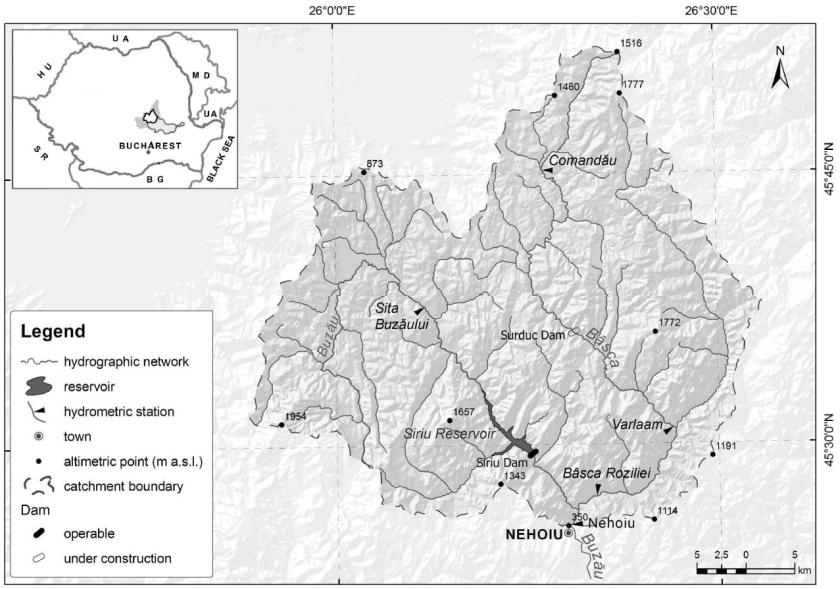

Figure 1. Buzău River basin in Romania [37].

Eighty percent of its annual volume is collected in the upper part, upstream of Nehoiu. The multi-annual and specific mean flow are 25.2 m^3/s, and 17 $L/s \cdot km^2$, respectively [39]. The Buzău River's complex arrangement includes the Buzau River upper course in the Siriu-Nehoiasu zone and that of its tributary, Bâsca Mare. Siriu Dam started to work on 1 January 1984 on the upper reach of the Buzău River. It has a length of 122 m and a height of 570 m, a maximum storage volume of 125 million m^3, and occupies a surface of 420 ha, draining 56.1% of the Buzău River catchment [40].

In the Siriu Dam section, the multiannual flow rate is 9.59 m^3/s, the maximum flow with 0.01% insurance is 2900 m^3/s, with 0.1% insurance is 1720 m^3/s, and with 1% insurance is 980 m^3/s. The accumulation must supply drinking and industrial water to settlements and industrial plants downstream with about 2.5 m^3/s, and water for irrigation for 50,000 ha [39]. Studies [37–39] showed the change in the river discharge regimen after the dam entered into operation on 1 January 1984.

Taking into account the importance of the Buzău River in the economy of the region, we considered it necessary to conduct a deeper investigation into the results provided using statistical methods and provide more evidence about the river discharge modification after January 1984, using a different approach (modeling, in this study).

The analyzed series consists of the monthly average discharge of the Buzău River recorded at the Nehoiu hydrometric station (45°25′29″ latitude and 26°18′27″ longitude) from January 1955 to December 2010 (Figure 2).

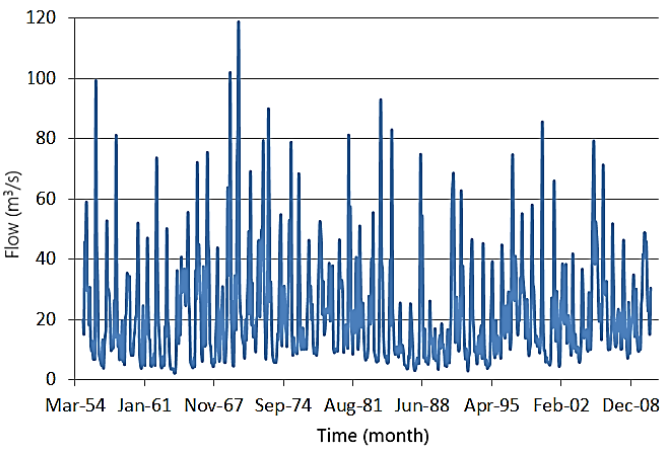

Figure 2. Monthly series of the Buzău River discharge.

The data series was automatically collected twice a day (at 7 a.m. and 7 p.m.) and transmitted to the National Institute of Hydrology and Water Management (INGHA), where they were verified by specialists who built the monthly average flow series from the daily data series. The series contains official data, without gaps, provided to us by INGHA for scientific purposes.

The basic statistics for S, S1, and S2 are presented in Table 1. S1 has the highest mean and variance indicating the highest variability of the river flow, confirmed by the existence of many flooding episodes before 1984. The lowest values of all statistics correspond to S2, showing a more homogenous distribution of the series values around the mean. All distributions are right-skewed and leptokurtic.

Table 1. The basic statistics and the results of the statistical tests (*p*-values).

	Minimum	Mean	Maximum	Variance	Coefficient of Variation (%)	Skewness	Kurtosis
S	2.18	21.83	117.29	306.82	80.23	1.79	3.93
S1	2.18	23.16	117.29	347.58	80.51	1.76	3.92
S2	2.93	20.41	92.79	259.14	78.87	1.76	3.43

The dataset was divided into a training set and a test set for the purposes of this study. The training set was different for each model: January 1955–September 2006 in Model S, January 1955–December 1983 (before putting the Siriu Dam in operation) in Model S1, and January 1984–September 2006 in Model S2 (after operating the dam). In all cases, the test set consists of data from October 2006 to December 2010.

2.2. Methodology

Classical approaches rely on the assumption of a constant data-generating process. They often fail to provide adequate models due to the nonlinear time series dynamic and the lack of adaptation of the method. Moreover, hydrological series are affected by permanently changing conditions, which are more or less abrupt. These issues make the river flow modeling problem well suited for utilizing ML approaches, which do not make any assumptions on the study process (assumptions generally imposed by other methods, like different regressions or Box–Jenkins methods).

Three alternative techniques are proposed here and described in the following paragraphs.

2.2.1. Convolutional Neural Networks-Long Short-Term Memory (CNN-LSTM)

CNN-LSTM [41] is a deep learning model that combines the characteristics of CNN [42] and LSTM [43,44] networks, designed for processing time-series data, image sequences, videos, and similar data types.

CNN is a deep learning model designed specifically for processing image data. It extracts features from images using convolutional layers and pooling layers. Convolutional layers employ convolution kernels to detect various features within an image while pooling layers reduce the dimensions of the feature maps. The mathematical representation of CNN is as follows:

- Convolution Layer Operation:

$$x^l = f^l\left(x^{(l-1)}\right) = \left(W^l * x^{(l-1)}\right) + b^{(l)}, \quad (1)$$

where $x^{(l-1)}$ is the feature map from the previous layer, W^l is the convolution kernel, $b^{(l)}$ is the bias term, and f^l is the activation function.

- Pooling Layer Operation:

$$x^l = g^{(l)}\left(x^{(l-1)}\right). \quad (2)$$

where $g^{(l)}$ is typically the maximum pooling or average pooling operation.

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer. The first layer is responsible for computing the dot product between the kernel (containing the parameters to be learnt) and the matrix containing the features map. The second layer has the role of reducing the representation size by processing individually its slice. The classification is performed in the third layer. It should be understood that "fully connected" expresses the connection between the inputs from one layer and all the nodes from the next layer.

LSTM [43] is a recurrent neural network designed for handling time-series data. It features memory cells that effectively capture long-term time dependencies. The mathematical representation of LSTM is as follows:

Input Gate: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$,

Forget Gate: $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$,

Candidate Unit: $\widetilde{C}_t = \tan h(W_C \cdot [h_{t-1}, x_t] + b_C)$,

Update Unit: $C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$,

Output Gate: $O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$,

Hidden State: $h_t = O_t * \tan h(C_t)$,

where t is the time (moment), x_t is the input at time step t, h_t is the hidden state at t, C_t is the cell state at t, i_t, f_t, and O_t are the forget and output gate units at t, W_i, W_f, W_C, W_o and b_i, b_f, b_C, b_o represent the weights and biases, respectively, at the input, forget, candidate, and output gates (Figure 3).

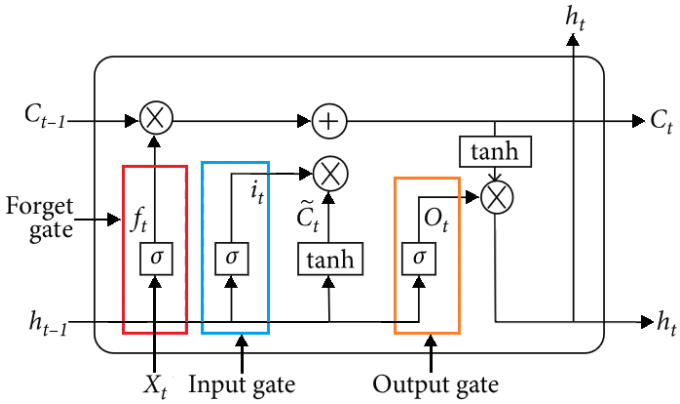

Figure 3. LSTM unit [45].

The LSTM can add or remove information to the cells and controls it by the gates (formed by a sigmoid neural net layer and a multiplication operation, *). The output of the sigmoid layer is zero (nothing will pass through the gate) or one (everything will pass). The Input Gate decides the values to be updated, and then the Candidate Unit builds the new candidates vector' s values, $\widetilde{C}_t$, updated in the Update Unit. The last two steps filter the information (in the Output Gate) that will exist in the network after applying a tanh function (to scale the values in the interval $[-1, 1]$) and multiplying by the sigmoid gate's output (in the Hidden state) [46].

In the CNN-LSTM approach (Figure 4), CNN is used to extract spatial features from the sequence data, while LSTM is employed to handle the temporal dependencies of these features.

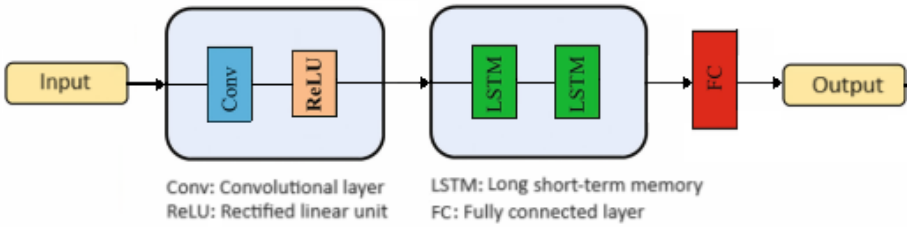

Figure 4. CNN-LSTM model.

The specific steps are as follows:

- CNN Processing: Input sequence data are processed through CNN to extract feature maps for each time step. These feature maps typically contain spatial information from the images.
- Sequence Processing: The feature maps from each time step are the inputs to LSTM. LSTM processes these feature maps and captures their temporal dependencies. The LSTM's hidden state is updated at each time step to capture long-term dependencies in the sequence.
- Output: The output from LSTM is used for time-series prediction.

CNNs are powerful for learning local patterns in data and feature extraction, while LSTMs are effective at capturing long-term dependencies in sequential data.

The benefit of CNN-LSTM is that the model can deal with very long input series, read as sub-series by the CNN and then combined by the LSTM model. Therefore, CNN will capture the data patterns, and LSTM will learn the temporal dependencies and make the final prediction. In such a way, improved performances of the combined model are obtained [47]. The new model can also extract nonlinear features and fluctuating trends [48].

2.2.2. Sparrow Search Algorithm-Backpropagation Neural Network (SSA-BP)

The sparrow search algorithm (SSA) is a novel population-based intelligent optimization algorithm inspired by the sparrows' foraging and anti-predatory behaviors. It was introduced in 2020 [49] and can be abstracted as the "Searcher-Follower" model with the inclusion of surveillance and warning mechanisms. The main actors in this algorithm are sparrows, with each sparrow having a single attribute: its position, representing the direction of discovered food. Sparrow individuals can undergo one of three types of state changes: (1) acting as searchers: leading the population to search for food; (2) becoming followers: following searchers in their food search; or (3) implementing surveillance and warning mechanisms: detecting danger and abandoning the food search.

In SSA, the best individuals within the population are prioritized for obtaining food during the search process. Explorers, as seekers, have a larger foraging search range than followers. During each iteration, the position update rule for explorers is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t+1} \cdot exp\left(\frac{-i}{\alpha \cdot iter_{max}}\right) & if\ R_2 < ST \\ X_{i,j}^{t} + Q \cdot L & if\ R_2 > ST \end{cases}, \tag{3}$$

where $X_{i,j}$ represents the position of a sparrow individual, i is the current iteration number, $iter_{max}$ is the maximum number of iterations, α is a random number in the range [0, 1], $R_2 \in [0, 1]$ and $ST \in [0.5, 1]$ are pre-alert and safety values, Q is a random number drawn from a normal distribution, and L is a $1 \times i$ matrix with all elements equal to 1. When $R_2 <$ ST, it signifies that there are no predators nearby, allowing explorers to conduct global searches. If $R_2 \geq ST$, it indicates that some sparrows have detected predators, and all sparrows need to take related actions. As previously mentioned, some followers constantly monitor the explorers during foraging. If explorers find better food, they will immediately leave their current location to compete for the food. If they win the competition, they can obtain the food instantly. The position update rule for followers is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot exp\left(\frac{X_{worst}^{t} - X_{i,j}^{t}}{i^2}\right), & i > \frac{n}{2} \\ X_{p}^{t+1} + \left|X_{i,j}^{t} - X_{p}^{t+1}\right| \cdot A^{+} \cdot L, & \text{otherwise.} \end{cases} \tag{4}$$

Here, X_p represents the position of the best explorer, X_{worst} is the current global worst position, and n is the population size. A is a $1 \times d$ matrix with each element randomly taking a value of 1 or -1. A^+ is defined by:

$$A^{+} = A^{T}\left(AA^{T}\right)^{-1} \tag{5}$$

When $i > n/2$, it means that the fitness of the i-th follower is relatively low, so they need to fly to other places to forage. In the algorithm, it is assumed that 10% to 20% of the individuals in the population become aware of the danger. These individuals' initial positions are randomly generated within the population:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta \cdot \left| X_{i,j}^{t} - X_{best}^{t} \right|, & if\ f_i > f_g \\ X_{i,j}^{t} + K \cdot \left(\frac{\left| X_{i,j}^{t} - X_{worst}^{t} \right|}{(f_i - f_w) + \varepsilon} \right), & if\ f_i = f_g \end{cases}, \quad (6)$$

where X_{best} represents the current global best position, β is a step size control parameter drawn from a normal distribution with a mean of 0 and a variance of 1, K is a random number in the range $[-1, 1]$, f represents the fitness value, f_g and f_w represent the current best and worst fitness values, and ε is a constant to avoid division by zero.

In summary, $f_i > f_g$ indicates that a sparrow is on the edge of the population, while $f_i = f_{sg}$ signifies that sparrows located in the middle of the population are aware of danger and need to move closer to other sparrows to avoid predation. K represents the direction of sparrow movement and is also a step size control parameter.

The backpropagation neural network (BPNN) is a commonly used supervised learning algorithm for solving classification and regression problems [50]. In the following, we present the detailed principle of the BPNN, including relevant formulas.

(1) Neurons and Activation Functions: The BPNN consists of multiple neurons, including input, hidden, and output layers. Each neuron has weights (w) and a bias (b). The neuron output is computed using an activation function (f), typically a sigmoid or ReLU function.

(2) Feedforward: Input elements from the input future vector X are passed through the input layer to the hidden and output layers, where each layer computes its output values. The output (O_j) of a layer is:

$$O_j = f\left(\sum_i w_{ij} \cdot X_i + b_j \right), \quad (7)$$

where w_{ij} is the weight for the node j for the incoming node i and b_j is the bias for the node j in the same layer.

(3) Training Data: The network is trained using labeled training data.

(4) Loss Function: The loss function measures the error between the model's output and the actual values (Y). Common loss functions include mean squared error (MSE) and cross-entropy loss.

(5) Backpropagation: The BPNN updates weights and biases to minimize the loss function using the backpropagation algorithm. It calculates the error term for the output layer, $E_j = \frac{1}{2}\left(Y_j - O_j\right)^2$, and uses the chain rule to compute the error term for the hidden layer, $E_h = f'(O_h) \cdot \sum_j w_{hj} \cdot E_j$. The weights and biases are updated as follows:

$$w_{ij}^{new} = w_{ij}^{old} + \eta \cdot E_j \cdot f'\left(O_j\right) \cdot X_i, \quad (8)$$

$$b_j^{new} = b_j^{old} + \eta \cdot E_j \cdot f'\left(O_j\right), \quad (9)$$

$$\mathrm{w_{hi}^{new}} = \mathrm{w_{hi}^{old}} + \eta \cdot \mathrm{E_h} \cdot \mathrm{f'(O_h)} \cdot \mathrm{X_i}, \quad (10)$$

$$b_h^{new} = b_h^{old} + \eta \cdot E_h \cdot f'(O_h), \quad (11)$$

Here, η is the learning rate that controls the step size for weight updates.

(6) Iteration: The feedforward and backpropagation steps are repeated until the loss function converges or reaches a predefined number of iterations.

(7) Output: Once training is completed, the BPNN can be used to predict the output for new input samples.

It is known that BP performance is greatly affected by the random selection of initial weights and thresholds. Due to its capability of exploring the global optimum in different search spaces and avoiding the problem of optimum local, SSA is used in determining the optimal weights and bias in the BP algorithm [51]. SSA-BP results by using SSA to optimize the objective function of BPNN and obtain the best parameters, followed by training and forecasting the series results.

2.2.3. Particle Swarm Optimization-Extreme Learning Machine (PSO-ELM)

Particle swarm optimization (PSO) [52] is an optimization algorithm inspired by collective behavior in birds or fish. The goal of PSO is to find the optimal solution by simulating the movement of individual particles in the solution space. Here is the detailed principle of the PSO algorithm, including relevant formulas:

(1) Initialization: Initialize the size of the particle swarm, N, and the position and velocity of each particle. Typically, each particle has a position vector X_i and a velocity vector V_i, representing the current position and velocity of the particle in the solution space.

(2) Compute Fitness: For each particle i, calculate its fitness value $f(X_i)$. The fitness function is the objective function to be optimized.

(3) Update Individual Best Position: For each particle i, update its individual best position P_i, which is the best known position. When $f(X_i)$ is better than $f(P_i)$, update P_i to X_i.

(4) Update Global Best Position: Select the global best position, P_g, as the best known global position from the individual best positions of all particles.

(5) Update Velocity and Position: For each particle i, update its velocity and position using the following equations:

$$V_i(t+1) = wV_i(t+1) + c_1r_1(P_i - X_i) + c_2r_2(P_g - X_i) \quad (12)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (13)$$

where t is the current iteration number, w is the inertia weight that controls the particle's inertia, c_1 and c_2 are learning factors, and r_1 and r_2 are random numbers introduced for randomness.

(6) Iteration: Repeat steps (2) to (5) until termination conditions are met, such as, for example, reaching the maximum number of iterations or finding a solution that meets convergence criteria.

(7) Output: Output the global best position P_g, which represents the discovered optimal solution.

The core idea of PSO is to explore the solution space by simulating the collective behavior of particles. Each particle updates its position and velocity based on its own experience and the global best position. The PSO's performance is influenced by parameters like w, c_1, and c_2, which require appropriate tuning for optimal performance. PSO is commonly used for solving optimization problems, especially in continuous and high-dimensional spaces.

The extreme learning machine (ELM) [53] is a fast neural network training algorithm for supervised learning tasks. Its core principles involve the initialization of a neural network and weight learning. ELM works based on the following principles:

(1) Initialization of the Neural Network:

Input Layer: ELM accepts input feature vectors x, typically represented by $x = [x_1, x_2, \ldots, x_d]$, where d is the feature dimension.

Hidden Layer: ELM initializes a random weight matrix W, usually represented as $W = [w_1, w_2, \ldots, w_M]$, where M is the number of hidden layer neurons. Weights are drawn from random distributions, such as uniform or Gaussian distributions.

(2) Hidden Layer Output: The output of the hidden layer, H, is calculated as $H = g(Wx + b)$, where $g(\cdot)$ is the activation function, typically sigmoid or ReLU, and b is the bias, usually set to zero.

(3) Output Layer Weight Learning: The key to ELM is the weight learning at the output layer, which can be achieved through the least squares method. For classification problems,

ELM's output is usually represented as $Y = [y_1, y_2, \ldots, y_C]$, where C is the number of classes. The output layer weight matrix is often denoted by $O = [o_1, o_2, \ldots, o_C]$.

The output layer weight O can be calculated using the following formula: $O = H^+T$, where H^+ is the Moore–Penrose pseudo-inverse of the hidden layer output matrix H, and T is the class label matrix, with each row corresponding to the class label of a sample.

(4) Prediction: Once the output layer weight O is determined, ELM can be used for forecasting new data. For a new input feature vector x_{new}, its predicted output y_{new} can be calculated using the formula:

$$y_{new} = g(W_{new}x_{new} + b)O, \quad (14)$$

where W_{new} represents the weights from the hidden layer to the output layer, specifically $W_{new} = H^+_{new}$. Moreover, H^+_{new} is the hidden layer output for the new data.

Huang et al. [54] demonstrated the ELM's ability to perform as a universal approximator. It was also shown [55,56] that ELM has a fast learning capability and adequate generalization performance, and combined with other techniques, can enhance its generalization ability [57,58]. However, due to the random initialization of the input weights, ELM may generate non-optimal solutions (affecting algorithm performance) [59,60].

To address this issue, PSO-ELM was applied in the following steps:

(a) Set the training and test set;
(b) Initialize the ELM parameters and set the root mean squared error (RMSE) as the fitness function;
(c) Run PSO for each candidate solution;
(d) Determine the optimal input data for ELM;
(e) ELM Test [32].

According to [32,61,62], PSO-ELM models provided highly reliable solutions to engineering problems. We mention that our scientific literature search did not return results on modeling water discharges using such an approach. Therefore, given that no reliable models for the Buzău River flow were found, we decided to apply this modeling technique.

2.2.4. Data Segmentation

The modality of dividing the data series into the training and testing datasets can also impact training effectiveness. Since the data being used include water flow data with an associated date, the optimal division method is based on the date. Additionally, the proportion of the testing dataset should be considered. Generally, a larger proportion of the training dataset may help the model learn time-series patterns more effectively. However, a smaller testing dataset may lead to inadequate evaluation of the model's performance. In time-series forecasting, a substantial amount of historical data is often necessary to build accurate models. Therefore, increasing the proportion of the training dataset might be beneficial, especially for long-term time-series data, to ensure the model has enough historical information to capture patterns within the time series. However, specific data characteristics and the available data quantity should also be considered. If the data are very limited, allocating more data for training may not be feasible. Furthermore, the size of the testing dataset should be sufficiently large to ensure a comprehensive evaluation of the model's performance. Ultimately, the appropriate ratio depends on experimental requirements and the available data.

Variables were standardized to compare the three models, and the data from January 2006 to December 2010 were designated as the testing dataset. Model S's training dataset encompasses data from January 1955 to December 2005. Model S1's training dataset comprises the period from January 1995 to December 1983. Model S2 is trained using data from January 1984 to December 2005. This approach aligns with the intent of this paper to determine the model with the best predictive performance of the test dataset, and emphasize the existence of a change in the water discharge regimen after 1984.

Table 2 contains the information on data segmentation.

Table 2. Data set segmentation—number of values per series set.

Model	Full Data Range (YYYYMM)	Training Data Range (YYYYMM)	Test Data Range (YYYYMM)	Test Set to Training Set Ratio
S	195501–201012 (672)	195501–200512 (612)	200601–201012 (60)	9.8%
S1	195501–198312 (348)	195501–198312 (348)	200601–201012 (60)	17.2%
S2	198401–201012 (324)	198401–200512 (264)	200601–201012 (60)	22.7%

2.2.5. Description of Algorithmic Running Parameters

This study employed three forecasting algorithms: CNN-LSTM, SSA-BP, and PSO-ELM. We conducted comprehensive parameter tuning and experiments to analyze if they demonstrate good predictive performance in practical applications.

For CNN-LSTM, a series of experiments were conducted to determine the optimal parameter configurations, including parameters for the convolutional layers and settings for the LSTM layers. This experimental process was instrumental in ensuring that CNN-LSTM achieved the best performance in handling flood data and feature extraction.

We applied the same parameter-tuning methodology to SSA-BP and PSO-ELM to ensure fairness in comparative results. The purpose of this consistent approach was to test the performance of these two algorithms under similar conditions, thus enhancing the credibility of the comparison. Through a similar parameter-tuning process, we optimized SSA-BP and PSO-ELM to achieve the best performance on the specific flood forecasting task and dataset.

This consistent parameter-tuning approach helps eliminate performance biases that could be introduced due to different parameter settings. Consequently, each algorithm was tested for performance under thoroughly optimized conditions, and our evaluation results better reflect their actual performance in real-world applications. This procedure ensures our research's scientific rigor and reliability, making our conclusions more compelling. After practical testing, the selected parameters for SSA-BP and PSO-ELM are listed in Table 3.

Table 3. Parameters of SSA-BP and PSO-ELM.

Algorithm	Lower Limit of Value	Upper Limit of Value	Population Size	Maximum Iterations	No. of Hidden Nodes
SSA-BP	−500	500	100	20	100
PSO-ELM	−1	1	100	50	300

The model's structure and parameters' settings of the CNN-LSTM network are as follows:

(1) Input Layer: The model begins with a sequence input layer with an input data structure of [1 1 1], representing input at a single time step.
(2) Sequence Folding Layer: This layer is responsible for serializing the input data for sequence data processing.
(3) Convolutional Layers: The model includes two convolutional layers, named conv_1 and conv_2. Both convolutional layers have a kernel size of [1 1], where conv_1 contains 16 feature maps and conv_2 contains 32 feature maps. These convolutional layers are used to extract features from serialized data, aiding the network in understanding patterns in the input data.
(4) Activation Layers: Following each convolutional layer, a ReLU activation layer (relu_1 and relu_2) is introduced to add non-linearity and enhance feature extraction.
(5) Sequence Unfolding Layer: This layer corresponds to the Sequence Folding Layer and is used for deserializing data for further processing.
(6) Fully Connected Layer: This layer flattens the data from a serialized format for processing by the fully connected layers.

(7) LSTM Layers: The model comprises two LSTM layers, named lstm and lstm2. LSTM (long short-term memory) layers are employed for handling sequence data, with lstm outputting a sequence and lstm2 outputting the result from the last time step.
(8) Fully Connected Layer: This layer (fc) receives the output from the LSTM layers and maps it to a single output node.
(9) Regression Layer: Finally, there is a regression layer responsible for outputting prediction results.

Here are some specific parameter settings:

- MaxEpochs: The maximum number of training epochs is set to 100.
- InitialLearnRate: The initial learning rate is set to 0.01.
- LearnRateSchedule: The learning rate schedule follows a "piecewise" strategy.
- LearnRateDropFactor: The learning rate drop factor is 0.1.
- LearnRateDropPeriod: The learning rate drop period is 80% of the maximum training epochs.
- Shuffle: The dataset is shuffled before each training iteration.
- Plots: Training progress is visualized during the training process.
- Verbose: Detailed information is not displayed during the training process.

The combination of these parameters and network layers is designed for efficient flood forecasting, with the model continually improving its performance over a certain number of training epochs. This model integrates CNN and LSTM networks to effectively handle sequential data, making it well suited for discharge forecasting tasks.

2.2.6. Performance Evaluation Criteria

The performance of the models was assessed using computation time, mean squared error (MSE), mean absolute error (MAE), and coefficient of determination for the training and test set (R^2).

(1) Mean squared error (MSE) measures the average of the squared errors between the model's predictions and the actual observations. A lower MSE indicates a better fit of the model to the observed data. The formula for calculating MSE is

$$\mathrm{MSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / n, \tag{15}$$

where n is the number of data points, y_i is the actual observation, and $\hat{y}_i$ is the model's prediction.

(2) The mean absolute error (MAE) measures the average of the absolute errors between the model's predictions and the actual observations. Unlike MSE, MAE does not consider the square of errors, making it less sensitive to large errors. The formula for calculating MAE is

$$\mathrm{MAE} = \sum_{i=1}^{n} |y_i - \hat{y}_i| / n. \tag{16}$$

(3) The coefficient of determination for the training set (R^2) represents the goodness of fit of the model to the training set data. Its value ranges from 0 to 1, with a higher value indicating a better fit of the model to the training data. The computation formula is

$$R^2 = 1 - SSR/SST, \tag{17}$$

where SSR is the sum of squared residuals, and SST is the total sum of squares.

(4) The coefficient of determination for the test set is similar to R^2 for the training set, and it is used to assess the model's fit to independent test data. It provides a performance metric for the model on new data.

In summary, computation time is used to evaluate the computational efficiency of the model. MSE and MAE are used to measure the model's prediction errors, while R^2 for both the training and test sets is used to assess the model's fit to the data. In general, all these metrics help evaluate the model's performance, ensuring that it accurately fits the training data and performs well on new data.

2.2.7. Computational Setup

The computations in this study were conducted on a workstation equipped with an AMD Ryzen 9 5900X 12-Core Processor CPU (3.70 GHz, 12 cores, 24 threads), 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU. The operating system used was Windows 11, and the programming environment included Matlab R2023a. All training was carried out on the CPU, and there was no need for the GPU to be involved in the computation.

2.2.8. Comparison of Hybrid Models with Other Models Used in Hydrological Modeling

For comparisons, two different types of models have been used, ARIMA and MLP. Brief information on the mentioned approaches is presented in the following paragraphs. We do not insist on them since they do not fall under the main goal of the article.

An ARIMA(p,d,q) model for a time series $\{x_t\}$ has the equation

$$\left(1 - \varphi_1 B - \ldots - \varphi_p B^p\right)\nabla^d x_t = \left(1 - \theta_1 B - \ldots - \theta_q B^q\right)\varepsilon_t,\ \varphi_p \neq 0,\ \theta_q \neq 0, \tag{18}$$

where

$$\nabla^d = (1 - B)^d, \tag{19}$$

p is the autoregressive order, q is the moving average order, d is the differentiation degree, and $\{\varepsilon_t\}$ is a white noise with zero mean and a constant variance [63].

The autoregressive moving average model of the orders p and q, denoted by ARMA(p, q) is an ARIMA(p, 0, q). If $p = d = 0$, the model is called moving average MA(q).

The best model is that with the lowest value of the Akaike criterion.

Multilayer perceptron (MLP) neural network [64] is a feedforward ANN formed by fully connected neurons organized in minimum three layer. In this article, we used four layers, two of which are hidden. In the actual MLP, an logistic activation function was used for both hidden and output layer. The classical and still preferred training algorithm for neural networks is called stochastic gradient descent. Network size evaluation was performed using a 20% hold-back procedure.

3. Results and Discussion

3.1. Modeling Results

Following the described methodology, we first modeled the data series, and then conducted a comprehensive performance comparison of three prediction approaches (CNN-LSTM, SSA-BP, and PSO-ELM) and the related models (S, S1, and S2) on the river discharge data series.

In the realm of computational modeling, particularly with algorithms that incorporate elements of randomness or stochastic processes, the role of the random seed is pivotal in determining the outcome of each run. In our study, we employed models which inherently involve random search methods in their optimization or learning processes. The random seed in these algorithms influences the initialization of weights, the selection of subsets of data, and the trajectory of the search process in the solution space.

Acknowledging the influence of random seeds, our methodology incorporated measures to ensure a fair and unbiased comparison across all iterations. To mitigate the variability introduced by random seeds, we adopted the following approaches:

- Multiple runs with different seeds: Each model was run 20 times with a range of different random seeds. This approach averages out the anomalies that might arise from any particular initialization, providing a more generalizable and reliable estimate of each model's performance.
- Consistent seeds across models: For each iteration, the same set of random seeds was used across all models. This consistency ensures that each model is subjected to the same degree of randomness in their respective processes, allowing for a fairer comparison of their capabilities.

Through these methodological choices and analytical approaches, we aimed to ensure that our comparison of the CNN-LSTM, SSA-BP, and PSO-ELM models was as fair and unbiased as possible. This rigorous approach allowed us to draw more reliable conclusions about the relative strengths and applicability of these models in the context of water discharge forecasting.

Figure 5 presents the visualization of the CNN-LSTM S, S1, and S2 models.

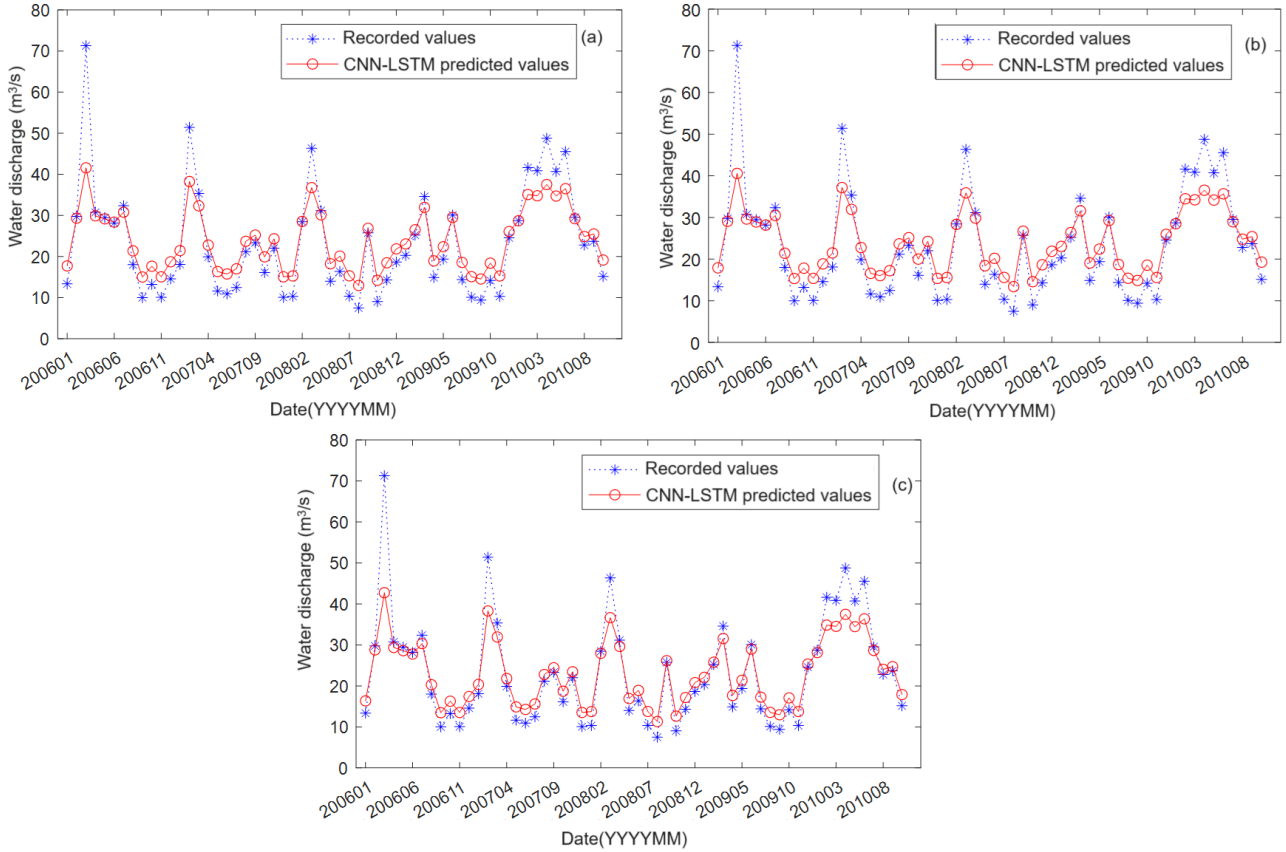

Figure 5. CNN-LSTM models on the test set when the training set was (**a**) S, (**b**) S1, and (**c**) S2.

The analysis of the three charts indicates that all models follow the shapes of the recorded data series. A higher bias of the computed values from the recorded ones is noticed for the extremes (for example, during the periods May–July 2007, September 2009, and March–June 2006) in S1 (Figure 5b) compared to S and S2 (Figure 5a,c). The lowest biases are noticed in Figure 5c—Model 2. Still, the differences are insignificant, and reflected in the goodness of fit indicators that will be discussed later in this section.

The output of the SSA-BP is displayed in Figure 6. Whereas the series formed by the computed values in the models S and S1 have similar shapes as the recorded series, with some higher biases between the recorded and computed values in S1 compared to S2, especially for the values from August and October 2008, and February–June 2010, the shape of the series in Model S2 is quite different. It is worth noting the mismatches between the recorded and forecasted values after February 2009, March and August 2008, etc., leading to the lowest performance of S2 compared with the S and S1 SSA-BP models.

The PSO-ELM output is represented in Figure 7. Similar to the previous approaches, the worst fitted are the highest values. For the values recorded in March 2006 (over 70 m^3/s), March 2007 (about 50 m^3/s), and March 2008 (46 m^3/s), and those after February 2010, the best estimations are provided by Model S2 (Figure 7c).

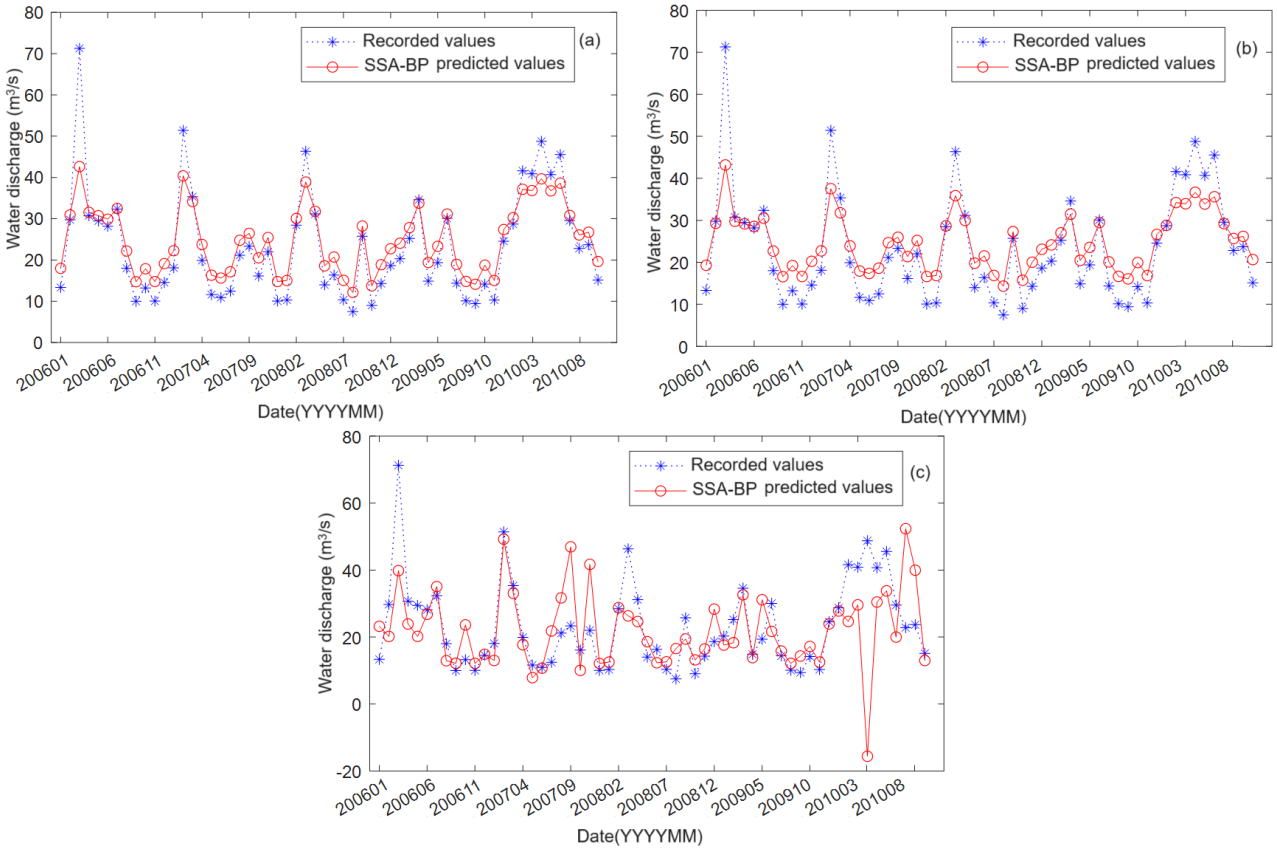

Figure 6. SSA-BP models on the test set when the training set was (**a**) S, (**b**) S1, and (**c**) S2.

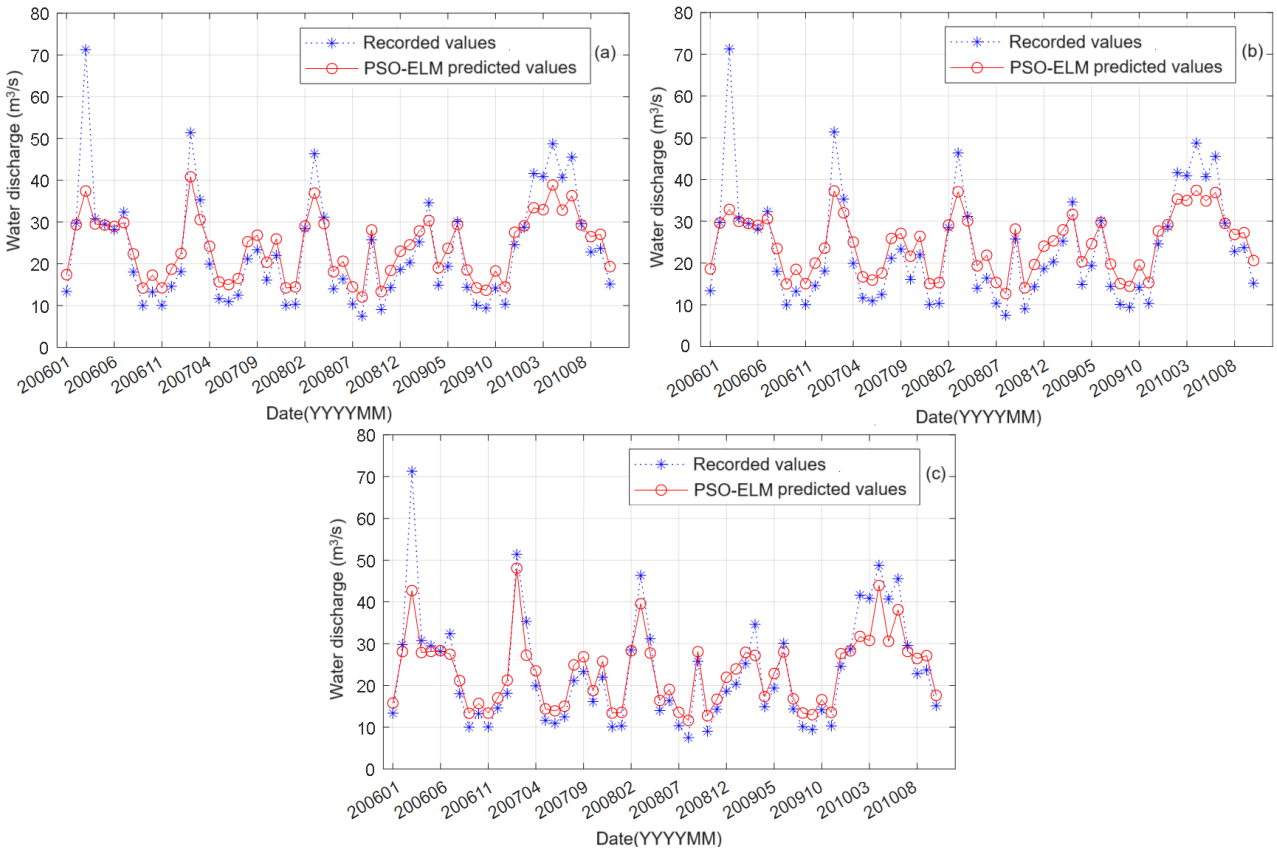

Figure 7. PSO-ELM models on the test set when the training set was (**a**) S, (**b**) S1, and (**c**) S2.

The same is true for the lowest recorded values, meaning that Model S2 better captures the extreme compared to S and S1.

A comparison of Figures 5c and 7c shows that the values provided by the CNN-LSTM-S2 model for the highest records are smaller than those issued from the PSO-ELM-S2 model, indicating smaller errors in the second case. Still, correct conclusions on the models' performances can be drawn only after observing the goodness-of-fit indicators.

MSE is a key indicator for assessing predictive accuracy. The study computed MSE for both the training and test sets (Table 4—rows 3–5).

Table 4. Values of the goodness-of-fit indicators for the training and test sets in the models.

Indicator	Model	Training Set			Test Set		
		S	S1	S2	S	S1	S2
MSE	CNN-LSTM	93.8144	115.0937	62.0042	36.0007	39.9782	29.8323
	SSA-BP	91.2629	105.403	132.454	32.4993	44.6227	168.5962
	PSO-ELM	98.125	126.5485	70.7001	41.2751	52.1818	30.9637
MAE	CNN-LSTM	6.0307	6.5177	4.7433	4.2351	4.4784	3.5245
	SSA-BP	5.7250	6.9987	7.7131	4.2882	5.2037	8.0949
	PSO-ELM	6.0070	6.7809	5.0355	4.6031	5.1284	3.9898
R^2	CNN-LSTM	0.8945	0.8839	0.9301	0.9458	0.9426	0.9504
	SSA-BP	0.8397	0.9276	0.5311	0.9297	0.9612	0.1976
	PSO-ELM	0.8305	0.7596	0.7966	0.8868	0.8335	0.8994

The analysis reveals that the training set MSE ranges from 62.0042 (for CNN-LSTM-S2) to 132.454 (for SSA-BP-S2), whereas the corresponding test set MSE ranges from 29.8323 (for CNN-LSTM-S2) to 168.5962 (for SSA-BP-S2).

Among the models, the lowest test set MSE occurs when using CNN-LSTM, particularly in Model S2, with a value of 29.8323, corresponding to the lowest training set MSE (62.0042). This finding suggests that CNN-LSTM exhibits low error rates and fits the actual observations excellently. In contrast, SSA-BP in Model S2 shows the highest MSE, indicating comparatively poorer predictive performance, emphasizing the significant variation in predictive accuracy among different models. Overall, the training set MSE is significantly higher than the test set MSE in all models but SSA-BP-S2, which might be attributed to the larger volume of training data.

Generally, the MAEs for the test set remain lower than for the training sets. MAE, known for its robustness, is less sensitive to outliers, as it solely considers the absolute value of errors. The training set's lowest MAE corresponds to the CNN-LSTM-S2 model (4.7433) and the highest to the SSA-BP-S2 model. In the case of the test sets, the lowest MAE was computed in the CNN-LSTM-S2 model (3.5245) and the highest in the SSA-BP S2 (8.0949). For SSA-BP-S2, the MAE's ranking on the test set is similar to the MSE's ranking on the same set. These results highlight the superior predictive performance of CNN-LSTM in terms of MAE. However, the poorest predictive performance is still unexpectedly observed in SSA-BP-S2, given that both test and training sets belong to the period after putting the Siriu Dam in operation.

Regarding R^2 on the training set (Table 4, rows 9–11), all three models but SSA-BP-S2 consistently exhibited relatively high R^2 values, showing their ability to effectively explain variance in the test data. On the test set, R^2 recorded values over 0.8335 for all but the SSA-BP models. CNN-LSTM achieved the highest R^2 in Model S2 on both the training and test sets.

Compared to its competitors, PSO-ELM displayed the lowest R^2 values on Models S and S1, whereas CNN-LSTM and SSA-BP consistently demonstrated higher R^2 values, over 0.8397 on the same models. The R^2 values of the SSA-BP on Model S2 were very low on the test and the training sets, rendering it practically unusable.

These results reflect the predictive accuracy of the CNN-LSTM and PSO-ELM models on the training dataset while revealing their adaptability to different datasets.

The following should be noted, given each training set's significance and relating it to the algorithms' prediction accuracy.

- Before running the algorithms, it was expected to obtain the best results for the S2 models because the training and test sets were recorded after operating the Siriu Dam. However, the second algorithm performed differently than expected, providing the worst S2 model compared to S and S1.
- It was also expected that the S models better fit the data compared to S1 given that the training set includes data from both periods (before and after January 1984), so with different flow regimens. This happened in terms of MAE and MSE for all models. In terms of R^2, the assertion is true for CNN-LSTM and PSO-ELM models.
- It was also expected that S1 has the worst performance because the training and test sets came from different periods. But SSA-BP S1 is the best in terms of R^2, compared to SSA-BP S and SSA S1.

The residuals' analysis rejects the autocorrelation hypothesis. Figure 8 shows the residuals' correlograms in CNN-LSTM S2, SSA-BP S2, and PSO-ELM S2 (with 95% confidence limits). The normality hypothesis was tested using the Anderson–Darling test [65]. Table 5 contains the associated p-values. At the significance level of 5%, the residuals' normality was rejected in all models. At a significance level of 1%, the normality hypothesis cannot be rejected in the CNN-LSTM models (because the p-values are higher than 0.01).

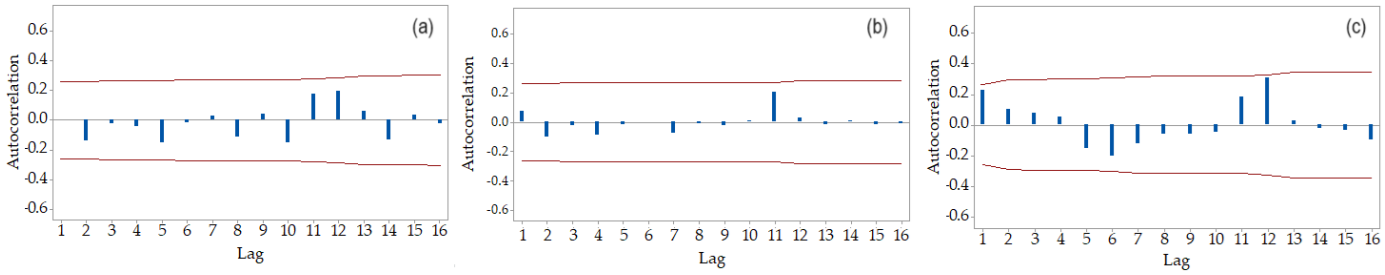

Figure 8. The correlograms of residuals in (**a**) CNN-LSTM S2, (**b**) SSA-BP S2, and (**c**) PSO-ELM S2. The vertical bars represent the values of the autocorrelation function, and the red lines are the limits of the confidence interval with 95% confidence limits.

Table 5. Analysis of residuals' normality in the models.

Model	CNN-LSTM		SSA-BP		PSO-ELM	
	p-Value	Normality Reached	*p*-Value	Normality Reached	*p*-Value	Normality Reached
S	0.018	Box-Cox: $\lambda = 1.22$	< 0.005	no	<0.005	no
S1	0.017	Box-Cox: $\lambda = 1.23$	< 0.005	no	<0.005	no
S2	0.031	Box-Cox: $\lambda = 1.15$	< 0.005	no	<0.005	no

Normality was reached by Box–Cox transformations [66], with the parameters 1.22 (1.23 and 1.15) for the residual series in the CNN-LSTM S (S1 and S2) models.

Considering the MSE, MAE, R^2, and the residuals' analysis, CNN-LSTM has the best performance. Its robustness, even in the presence of outliers, suggests a consistent predictive reliability. Additionally, the model's higher determination coefficient (R^2) values, both in the training and testing phases, indicate its enhanced capability to explain the variance in the dataset.

The complexity of hydrological patterns post Siriu Dam construction (including altered flow regimes and seasonal variations) are effectively captured by the CNN-LSTM model due to its architecture, which combines convolutional layers (for spatial feature recognition) and LSTM layers (for temporal dependencies), particularly adequate at modeling

non-linearity and non-stationarity. The robustness of CNN-LSTM in variable hydrological conditions is attributable to the LSTM component's ability to remember long-term dependencies and disregard irrelevant data. Moreover, the hierarchical patch-based convolution operations performed by CNNs reduce the computational effort, and the input is abstracted on different feature levels, diminishing the network's parameter number. Also, convolution layers consider the context in the local neighborhood of the input data and construct features from that neighborhood.

The performance of SSA-BP is relatively less impressive than those of other models. This behavior might be related to SSA's large randomness issue, easily falling into the local optimum. Moreover, the poor communication mechanism between the participants (that communicate only with the best discoverers) can result in missing the best solutions, affecting fitting quality [67].

ELM has good generalization and learning capacity (thousands of times faster than learning algorithms for feed-forward NN) [53]. PSO has a strong global exploration ability. It approaches the optimum solution by self- and social learning, continuously updating the global and historical optimal solutions. Therefore, the PSO-ELM will benefit from these characteristics and improve the PSO convergence rate.

3.2. Sensitivity Analysis

First, we have to mention that the models were built in scenarios S–S2 (described in Table 2) in order to determine if there is an alteration of the Buzău River flow after building the dam. Since sensitivity analysis is a very complex task for the complex ML algorithms used here, we decided to perform an extended analysis in another article.

But here we used the rolling origin evaluation technique according to which the forecasting origin is updated successively and the forecasts are produced from each origin [68]. In the case of ML techniques, this involves changing the ratio between the training and test sets.

Performing this analysis for all network types resulted in the highest sensitivity of the SSA-BP model, for which R^2 drastically decreased, whereas MSE increased, especially when the ratio test/training test is over 35%. For example, for a ratio of 38% (68%), MSE = 1020.7045 (2472.2555) on the training set and 2010.9544 (4291.441) on the test set. The corresponding R^2 decreased at values under 0.05 in the same cases, whereas for ratios under 22%, it remained around 0.828 on the training and 0.928 on the test set.

PSO-ELM had comparative performances with S1-S3 models on the training set in terms of all goodness-of-fit parameters, whereas on the test set, MSE and MAE slightly increased.

CNN-LSTM had almost the same values of R^2 as in the S–S2 scenarios. A moderate increase of MSE and a slight increase of MAE (in the range of 4.60 and 5.50) on the test sets were also noticed. Overall, the least sensitive model was CNN-LSTM.

3.3. Computational Time Complexity

The time needed to run the algorithms is also a crucial factor. It is presented in Figure 9 for each model as a function of the data volume on the training set. In our comprehensive analysis of computational time for the CNN-LSTM, SSA-BP, and PSO-ELM models in water discharge forecasting, distinct patterns emerged, highlighting the varying efficiencies of these models in handling datasets of different sizes.

The PSO-ELM model demonstrated a significant reduction in computational time as the data volume was diminished from 612 to 264, with a notable decrease by approximately a third when transitioning from Model S (612 data points) to Model S1 (348 data points). This augmentation of the computational burden with larger datasets indicates that PSO-ELM may not be optimally suited for scenarios involving extensive data, owing to its intrinsic algorithmic complexity that scales unfavorably with increased data dimensions and search space.

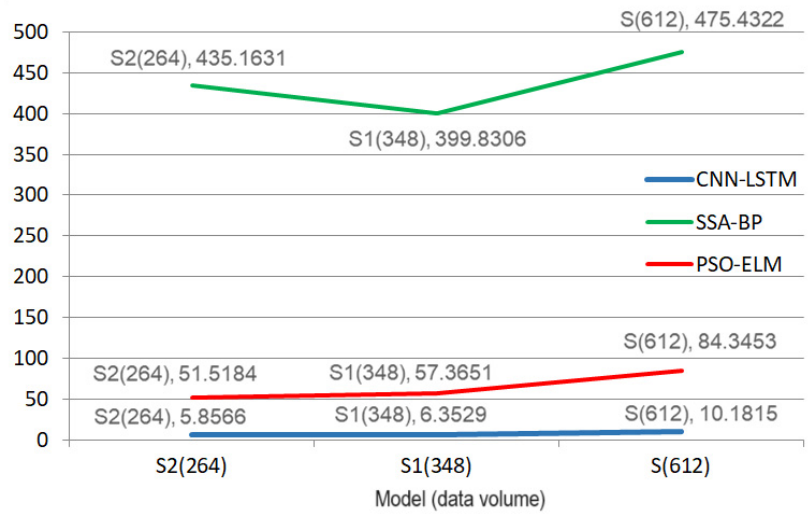

Figure 9. Computational time function of the number of values.

The CNN-LSTM model exhibited a near-linear relationship between data volume and computational time. This scalability, presumably a result of the parallelizable nature of CNNs for spatial feature processing and the linear time complexity of LSTMs with respect to sequence length, suggests its suitability for larger, more complex time-series datasets. Such a characteristic is particularly advantageous in real-time or near-real-time forecasting systems where handling extensive hydrological data efficiently is crucial.

The SSA-BP model, however, did not show a clear correlation between data volume and computational time, indicating that other factors than data volume, such as algorithmic structure, initialization parameters, and convergence criteria, play a more significant role in influencing its computational efficiency. This observation underscores the need for meticulous parameter optimization and algorithmic adjustments to harness the full potential of SSA-BP in specific hydrological forecasting scenarios.

Despite SSA's known fast convergence capacity, the very slow convergence of the BPNN impacted the total computational time of the SSA-BP algorithm [67,69].

Our comprehensive analysis reveals that the CNN-LSTM model exhibits exceptional performance, outshining its counterparts, SSA-BP and PSO-ELM, in several critical aspects. Firstly, the CNN-LSTM model demonstrates a marked efficiency in computational time, leveraging GPU acceleration to process extensive datasets rapidly, as evidenced by its remarkable processing time of only 5.8566 s in Model S2. This efficiency is crucial when dealing with decade-spanning hydrological data, as in our study.

- *CNN-LSTM time complexity.*

The convolutional layer complexity time in CNN is $O\left(\sum_{l=1}^{k} n_{l-1} s_l^2 n_l m_l^2\right)$ [70], where k is the number of convolutional layers (two in our case), n_l is the number of filters in the l-th layer, n_{l-1} is the number of input channels of the l-th layer, s_l is the spatial size of the filter, and m_l is the spatial size of the output feature map [71]. Hochreiter and Schmidhuber proved [43] that the LSTM is local in space and time, so the time complexity per weight for each time step is O (1). Therefore, the overall complexity of an LSTM per time step is equal to O (w), where w is the number of weights. Therefore the CNN-LSTM complexity per time step is $O\left(\sum_{l=1}^{k} n_{l-1} s_l^2 n_l m_l^2 + w\right)$, and for the entire training process is $O\left(\left(\sum_{l=1}^{k} n_{l-1} s_l^2 n_l m_l^2 + w\right) NM\right)$, where N is the input volume and M is the number of iterations [72].

- *PSO-ELM time complexity:*

In a PSO:

(1) If N particles are initialized and the solution space has the dimension d, the time complexity is O(Nd).

(2) The time complexity for the fitness function is O(d). The time complexity of the fitness computation for all the n particles is O(Nd).
(3) The time complexity in an iterative operation for updating and computing the extremum for each particle is O(N).
(4) The time complexity for the global extremum computation in an iterative operation is O(N).
(5) The time complexity in an iterative operation to update the velocity and position vectors of the particles is O(Nd).
(6) The time complexity of completing the computation after an iterative step, according to the termination condition, is O(1).

Summing up the time complexities in (2)–(6) results in O($2Nd + 2N + 1$), so the time measure level is O(Nd) [73]. If the algorithm runs M times, the complexity is O($2MNd + 2NM + M$), and the entire time complexity is O($2MNd + Nd + 2NM + M$), so the measure level is O(MNd).

In an ELM that transforms the input feature matrix with the dimension $d \times N$ to a hidden layer of h neurons, the following complexities for each computational step are determined:

(a) Linear transform to ELM-space-O(hdN).
(b) Application of activation function, assuming ReLU-O(Nh).
(c) Calculating the output weight matrix-O(N^3).

Adding up the values from (a)–(c) results in the ELM time complexity being O($hdN + Nh + N^3$), so the measure level is O(N^3) [74].

Based on the above, the PSO-ELM algorithm's time complexity is O($2MNd + Nd + 2NM + M + hdN + Nh + N^3$), so the measure level is O($MNd + N^3$).

- *SSA-BP time complexity*

The time complexity of the BPNN is influenced by the maximum number of iterations M, the sample size N, and the spatial dimension d, and its time complexity is O($M\,Nd^2$). After using the SSA algorithm with BPNN, the complexity increases by O(MNd). Thus, the time complexity of the SSA-BP algorithm is O($Mnd^2 + Mnd$) [75].

When the spatial dimension d is high, O(MNd^2) and O(MNd) are approximated by O(d^2), so, in such a case, SSA-BP has the complexity in the same measure level, O(d^2).

These insights into the models' computational time complexities have profound implications for their application in practical scenarios. While PSO-ELM may be more suited for smaller datasets or situations where longer computation times are acceptable, CNN-LSTM, with its excellent scalability and linear computational time relationship, emerges as a more viable option for applications demanding the rapid processing of large-scale data sets, such as dynamic hydrological models or real-time prediction systems. The SSA-BP model, requiring careful tuning and optimization, could be effectively employed in specific scenarios, provided that its parameters are optimally adjusted to the unique demands of the task at hand. This analysis not only aids in selecting the most appropriate model for a given hydrological forecasting application but also contributes to the broader understanding of leveraging advanced computational methods in environmental science research.

3.4. Discussion

Among the most used techniques for modeling river discharge, ARIMA and ANN are the most well known. Therefore, to compare the output of the hybrid models proposed in this article with the results from the literature, we built ARIMA-type models, denoted S_A, S1_A, and S2_A, with the series used for training the models S, S1, and S2 in the hybrid algorithm. Based on each model, the forecast was performed for the next 60 months and compared to the test set from the hybrid models. The types and the coefficients of the best ARIMA models are presented in Table 6.

Table 6. The coefficients (and standard errors—s.e.), MSE, MAE, and R^2 in the ARMA and MA models.

Model	Type	ar1 (s.e.)	ar2 (s.e.)	ar3 (s.e.)	ma1 (s.e.)	ma2 (s.e.)	Mean (s.e.)	MSE	MAE	R^2
S_A	ARMA(3, 1)	1.0310 (0.1106)	−0.1852 (0.0804)	−0.1500 (0.0416)	−0.5562 0.1066		21.6297 0.9047	234.7931	10.9039	0.2680
S1_A	ARMA(3, 1)	0.9942 (0.1166)	−0.1272 (0.0917)	−0.2089 0.0544	−0.5803 0.1108		23.1170 1.0657	260.6423	11.4934	0.2501
S2_A	MA(2)				0.4375 0.0572	0.2935 0.0606	23.1397 1.8383	299.0880	12.3642	0.4882

First, the portmanteau tests (Box–Ljung and Box–Pierce tests) [63] applied to the models' residuals rejected the autocorrelation of the residuals. The MSEs and MAEs (R^2) in the ARMA and MA models are generally much higher (lower) than those on the hybrid models, indicating an inferior output accuracy.

The forecast based on S_A, S1_A, and S2_A is shown in Figure 10, compared with the recorded series values. The forecast series becomes linear after a short period. Thus, the models fail to capture the nonlinearities in the recorded series. By comparison, the MLP models' fits in similar scenarios are better than the ARIMA ones, given their capacity to capture the abrupt changes in the series behavior. Figure 11 presents the modeling results using MLP in the first scenario for exemplification.

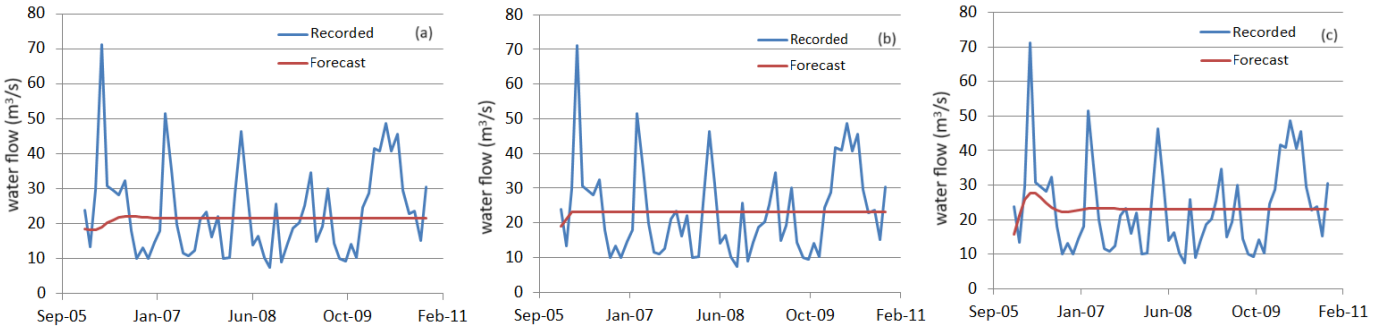

Figure 10. Recorded values and forecast by the (**a**) S_A, (**b**) S1_A, and (**c**) S2_A models.

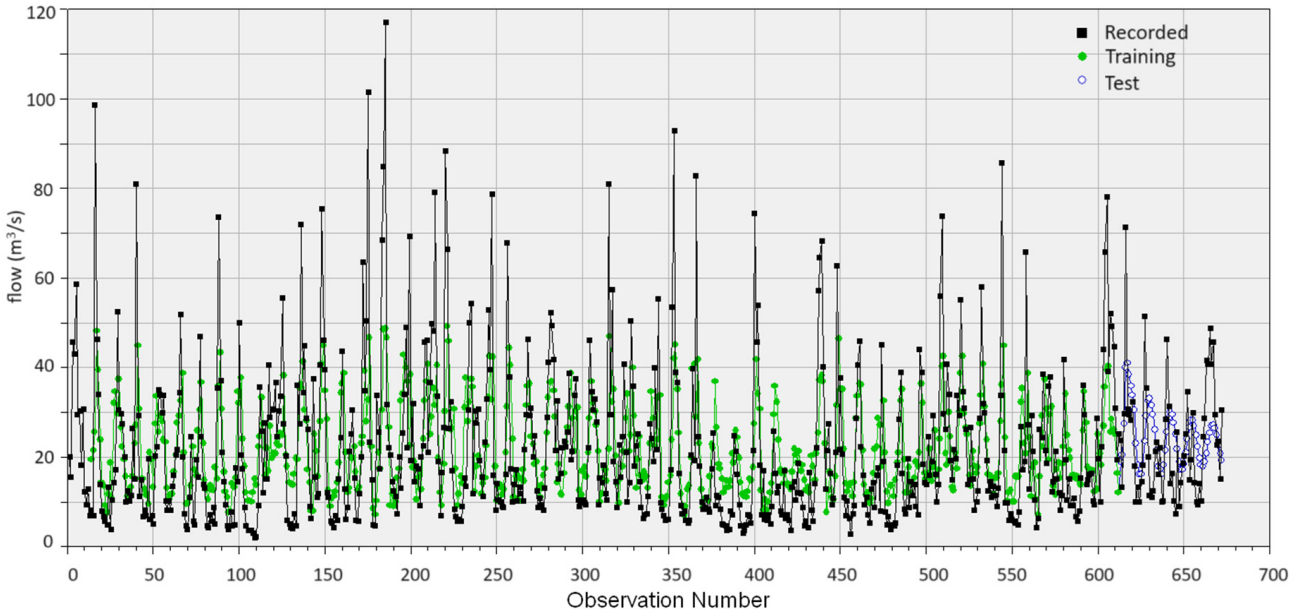

Figure 11. MLP models in S_A.

The forecast values on the test series (the points in blue) better follow the pattern of the recorded ones (the black dots).

The results of our ARIMA models are not in concordance with the findings of Yürekli et al. [15], whose simulations describe the recorded series well. They do not confirm the similar forecasting power of ANN and ARMA found by Abrahart and See [18] in a case study from Turkey. Our study output is in concordance with the findings from [17,76,77] that emphasized the better performances of the neural networks against ARIMA, given the better ability to learn of the neural network and less sensitivity to the abrupt changes in the time series.

The variation intervals for the goodness-of-fit indicators in the MLP models are as follows:

- On the training set: MSE—181.6991 (S2)–252.1943 (S1), MAE—9.2589 (S2)–11.4146, (S1), and R^2—0.2744 (S1)–0.3693 (S2);
- On the test set: MSE—135.7250 (S1)–158.1449 (S2), MAE—8.8535 (S1)–10.1030 (S2) and R^2—0.0969 (S)–0.1618 (S1).

So, MLP S1 performed the worst on the training set (it has the lowest R^2), and the highest and the best on the test one was S1. Still, the low R^2 indicates the necessity of running many optimization scenarios to overcome MLP's known drawback which is the difficult parameters' optimization [78]. Since the goal of this article is not modeling with MLP, we leave it for a further deeper research.

The performances of the MLP algorithm in all scenarios are worse than those of the hybrid models. The results regarding the predictive performance of both MLP and CNN-LSTM concord with those from [45,79,80]. They were somehow expected, given the advantages of CNN (smaller weights, shared, easy to train, going deeper, sparsely connected layers) over MLP, which are shared by the CNN-LSTM network. Moreover, the high forecast accuracy of CNN-LSTM, emphasized by this study, confirms its high performance in the case of long data series [81]. The results are also in concordance with those of Liu et al. [82] and Anupan and Pani [83], who indicated the considerable accuracy of the PSO-ELM network in terms of MSE and MAE, even when forecasting is carried out for a long period (60 months, in this case).

4. Conclusions

Against the backdrop of rapid technological advancement, hybrid computational methods have emerged as key tools for solving complex problems. These methods, integrating the strengths of various algorithms, offer more efficient and precise solutions for specific challenges. This paper rigorously examined the efficacy of three such hybrid models—CNN-LSTM, SSA-BP, and PSO-ELM—in the context of water discharge forecasting for the Buzău River, particularly in the wake of environmental changes induced by the Siriu Dam's operationalization in 1984.

Through a comprehensive analysis of runtime, MSE, and R^2, it can be concluded that CNN-LSTM and PSO-ELM can be used with good results on various cases (training sets) from flow forecast. CNN-LSTM stands out due to its computational efficiency and high predictive accuracy, especially in the case of Model S2. Its robustness extended to MAE, emphasizing CNN-LSTM's consistency, even in the presence of outliers.

We found that computational time was a crucial consideration, with CNN-LSTM demonstrating a significant advantage due to its efficient GPU utilization. It excelled in Model S2, requiring only 5.8566 s for processing, while SSA-BP and PSO-ELM, running on CPUs, consumed considerably more time. Therefore, for practical flood prediction, Model S2 is recommended as the primary choice due to its short runtime, low MSE, reasonable MAE, and test and training set R^2 values exceeding 0.92, indicating excellent fit without overfitting.

Regarding the determination coefficient (R^2), CNN-LSTM and SSA-BP consistently showed higher values, indicating their better ability to explain the variance in test data. In contrast, PSO-ELM exhibited relatively lower R^2 values, hovering around 0.83 to 0.9,

suggesting its diminished performance under certain circumstances. The training set R^2 mirrored these trends, with CNN-LSTM achieving the highest values in Model S2 and SSA-BP the lowest in Model S1. It should be noted that SSA-BP S1 performed best on both training and test sets in terms of R^2.

The results of the study confirm the existence of a different behavior in the monthly river discharge of the Buzău River, emphasized by the lowest performances of the models built using as training and test sets the series before, and after January 1984, respectively, i.e., Model S1.

In conclusion, the CNN-LSTM model's advanced architectural design, coupled with its ability to efficiently process large datasets and adapt to significant environmental changes, positions it as a highly effective tool for water discharge prediction in altered river systems. This study not only underscores the model's potential for widespread application in hydrological research but also offers invaluable insights for the scientific community and policymakers in enhancing our understanding and management of global water resources in an era marked by rapid environmental transformations.

Author Contributions: Conceptualization, L.Z. and A.B.; methodology, L.Z. and A.B.; software, L.Z.; validation, L.Z. and A.B.; formal analysis, L.Z. and A.B.; investigation, L.Z. and A.B.; resources, A.B.; data curation, A.B.; writing—original draft preparation, L.Z. and 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 APC was funded in part by Transilvania University of Brașov, Romania.

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

Conflicts of Interest: The present research work carries no conflicts of interest.

References

1. Naiman, R.J.; Decamps, H.; Pollock, M. The role of riparian corridors in maintaining regional biodiversity. *Ecol. Appl.* **1993**, *3*, 209–212. [CrossRef]
2. Magilligan, F.J.; Nislow, K.H. Changes in hydrologic regime by dams. *Geomorphology* **2005**, *71*, 61–78. [CrossRef]
3. Popescu, C.; Bărbulescu, A. On the Flash Flood Susceptibility and Accessibility in the Vărbilău Catchment (Romania). *Rom. J. Phys.* **2022**, *67*, 811.
4. Dumitriu, C.S.; Bărbulescu, A.; Maftei, C. IrrigTool—A New Tool for Determining the Irrigation Rate Based on Evapotranspiration Estimated by the Thornthwaite Equation. *Water* **2022**, *14*, 2399. [CrossRef]
5. Bărbulescu, A.; Maftei, C. Statistical approach of the behavior of Hamcearca River (Romania). *Rom. Rep. Phys.* **2021**, *73*, 703.
6. Bucurica, I.-A.; Dulama, I.-D.; Radulescu, C.; Banica, A.L. Surface Water Quality Assessment Using Electroanalytical Methods and Inductively Coupled Plasma Mass Spectrometry (ICP-MS). *Rom. J. Phys.* **2022**, *67*, 802.
7. Chilian, A.; Tanase, N.-M.; Popescu, I.V.; Radulescu, C.; Bancuta, O.-R.; Bancuta, I. Long-Term Monitoring of the Heavy Metals Content (Cu, Ni, Zn, Cd, Pb) in Wastewater Before and after the Treatment Process by Spectrometric Methods of Atomic Absorption (FAAS and ETAAS). *Rom. J. Phys.* **2022**, *67*, 804.
8. Bhowmik, R.D.; Sankarasubramanian, A.; Sinha, T.; Patskoski, J.; Mahinthakumar, G.; Kunkel, K.E. Multivariate downscaling approach preserving cross correlations across climate variables for projecting hydrologic fluxes. *J. Hydrometeorol.* **2017**, *18*, 2187–2205. [CrossRef]
9. Vrac, M.; Thao, S.; Yiou, P. Changes in temperature–precipitation correlations over Europe: Are climate models reliable? *Clim. Dyn.* **2023**, *60*, 2713–2733. [CrossRef]
10. Dekens, L.; Parey, S.; Grandjacques, M.; Dacunha-Castelle, D. Multivariate distribution correction of climate model outputs: A generalization of quantile mapping approaches. *Environmetrics* **2017**, *28*, e2454. [CrossRef]
11. Bărbulescu, A.; Dumitriu, C.S.; Maftei, C. On the Probable Maximum Precipitation Method. *Rom. J. Phys.* **2022**, *67*, 801.
12. Rahayu, W.S.; Juwono, P.T.; Soetopo, W. Discharge prediction of Amprong river using the ARIMA (autoregressive integrated moving average) model. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *437*, 012032. [CrossRef]
13. Ghimire, B.N. Application of ARIMA Model for River Discharges Analysis. *J. Nepal Phys. Soc.* **2017**, *4*, 27–32. [CrossRef]
14. Valipour, M. Long-term runoff study using SARIMA and ARIMA models in the United States. *Meteorol. Appl.* **2015**, *22*, 592–598. [CrossRef]
15. 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]
16. MA Models for Forecasting: Pros, Cons, and Examples. Available online: https://www.linkedin.com/advice/0/what-advantages-disadvantages-using-arima (accessed on 4 January 2024).

17. Zhou, J.; Wang, D.; Band, S.S.; Jun, C.; Bateni, S.M.; Moslehpour, M.; Pai, H.-T.; Hsu, C.-C.; Ameri, R. Monthly River Discharge Forecasting Using Hybrid Models Based on Extreme Gradient Boosting Coupled with Wavelet Theory and Lévy–Jaya Optimization Algorithm. *Water Resour. Manag.* **2023**, *37*, 3953–3972. [CrossRef]
18. Abrahart, 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]
19. Birikundavyi, S.; Labib, R.; Trung, H.T.; Rousselle, J. Performance of Neural Networks in Daily Streamflow Forecasting. *J. Hydrol. Eng.* **2002**, *7*, 392. [CrossRef]
20. Hong, J.L.; Hong, K. Flood Forecasting for Klang River at Kuala Lumpur using Artificial Neural Networks. *Int. J. Hybrid Inf. Technol.* **2016**, *9*, 39–60. [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. Techn.* **2015**, *4*, 184–188.
23. 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]
24. 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]
25. Li, S.; Yang, J. Modelling of suspended sediment load by Bayesian optimized machine learning methods with seasonal adjustment. *Eng. Appl. Comput. Fluid Mech.* **2022**, *16*, 1883–1901. [CrossRef]
26. Hayder, G.; Solihin, M.I.; Mustafa, H.M. Modelling of River Flow Using Particle Swarm Optimized Cascade-Forward Neural Networks: A Case Study of Kelantan River in Malaysia. *Appl. Sci.* **2020**, *10*, 8670. [CrossRef]
27. Xiang, X.J.; Li, Q.; Khan, S.; Khalaf, O.I. Urban water resource management for sustainable environment planning using artificial intelligence techniques. *Environ. Impact Assess. Rev.* **2021**, *86*, 106515. [CrossRef]
28. Essien, A.E.; Chukwukelu, G.; Giannetti, C. A Scalable Deep Convolutional LSTM Neural Network for Large-Scale Urban Traffic Flow Prediction using Recurrence Plots. In Proceedings of the 2019 IEEE Africon, Accra, Ghana, 25–27 September 2019; pp. 1–7.
29. Zhang, J.X.; Li, S.Y. Air quality index forecast in Beijing based on CNN-LSTM multi-mode. *Chemosphere* **2022**, *308*, 136180. [CrossRef]
30. Yan, P.; Shang, S.; Zhang, C.; Yin, N.; Zhang, X.; Yang, G.; Zhang, Z.; Sun, Q. Research on the Processing of Coal Mine Water Source Data by Optimizing BP Neural Network Algorithm with Sparrow Search Algorithm. *IEEE Access* **2021**, *9*, 108718–108730. [CrossRef]
31. Xin, J.X.; Chen, J.Z.; Li, C.Y.; Lu, R.K.; Li, X.L.; Wang, C.X.; Zhu, H.W.; He, R.Y. Deformation characterization of oil and gas pipeline by ACM technique based on SSA-BP neural network model. *Measurement* **2022**, *189*, 110654. [CrossRef]
32. Kaloop, M.R.; Kumar, D.; Samui, P.; Gabr, A.R.; Hu, J.W.; Jin, X.; Roy, B. Particle Swarm Optimization Algorithm-Extreme Learning Machine (PSO-ELM) Model for Predicting Resilient Modulus of Stabilized Aggregate Bases. *Appl. Sci.* **2019**, *9*, 3221. [CrossRef]
33. 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]
34. 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]
35. 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]
36. Mocanu-Vargancsik, C.A.; Bărbulescu, A. On the variability of a river water flow, under seasonal conditions. Case study. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *344*, 012028. [CrossRef]
37. Minea, G.; Bărbulescu, A. Statistical assessing of hydrological alteration of Buzău River induced by Siriu dam (Romania). *Forum Geogr.* **2014**, *13*, 50–58. [CrossRef]
38. Bărbulescu, A. Statistical Assessment and Model for a River Flow under Variable Conditions. Available online: https://cest2017.gnest.org/sites/default/files/presentation_file_list/cest2017_00715_poster_paper.pdf (accessed on 28 December 2023).
39. 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)
40. Chendeş, V. *Water Resources in Curvature Subcarpathians. Geospatial Assessments*; Editura Academiei Române: Bucureşti, Romania, 2011. (In Romanian with English Abstract)
41. Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Li, Z. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 5668–5675. [CrossRef]
42. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2006.
43. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
44. Gers, F. Long Short-Term Memory in Recurrent Neural Networks. Ph.D. Thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2001. Available online: http://www.felixgers.de/papers/phd.pdf (accessed on 29 November 2023).
45. Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-Based Model to Forecast Stock Prices 2020. *Complexity* **2020**, *2020*, 6622927. [CrossRef]

46. Colah's Blog. Understanding LSTM Networks. 2015. Available online: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 29 November 2023).
47. Aksan, F.; Li, Y.; Suresh, V.; Janik, P. CNN-LSTM vs. LSTM-CNN to Predict Power Flow Direction: A Case Study of the High-Voltage Subnet of Northeast Germany. *Sensors* **2023**, *23*, 901. [CrossRef]
48. Zhang, F.; Deng, S.; Wang, S.; Sun, H. Convolutional neural network long short-term memory deep learning model for sonic well log generation for brittleness evaluation. *Interpretation* **2022**, *10*, T367–T378. [CrossRef]
49. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
50. Rumelhart, D.; Hinton, G.; Williams, R. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
51. Wang, X.; Liu, J.; Hou, T.; Pan, C. The SSA-BP-based potential threat prediction for aerial target considering commander emotion. *Defen. Techn.* **2022**, *18*, 2097–2106. [CrossRef]
52. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell* **2007**, *1*, 33–57. [CrossRef]
53. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
54. Huang, G.-B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [CrossRef]
55. Karami, H.; Karimi, S.; Bonakdari, H.; Shamshirband, S. Predicting discharge coefficient of triangular labyrinth weir using extreme learning machine, artificial neural network and genetic programming. *Neural Comput. Appl.* **2018**, *29*, 983–989. [CrossRef]
56. Cui, D.; Bin Huang, G.; Liu, T. ELM based smile detection using Distance Vector. *Pattern Recognit.* **2018**, *79*, 356–369. [CrossRef]
57. 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. Agric.* **2020**, *173*, 105430. [CrossRef]
58. 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]
59. Cao, J.; Lin, Z.; Huang, G.B. Self-adaptive evolutionary extreme learning machine. *Neural Process. Lett.* **2012**, *36*, 285–305. [CrossRef]
60. Mohapatra, P.; Chakravarty, S.; Dash, P.K. An improved cuckoo search based extreme learning machine for medical data classification. *Swarm Evol. Comput.* **2015**, *24*, 25–49. [CrossRef]
61. Chen, S.; Shang, Y.; Wu, M. Application of PSO-ELM in electronic system fault diagnosis. In Proceedings of the 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Ottawa, ON, Canada, 20–22 June 2016.
62. Liu, D.; Li, G.; Fu, Q.; Li, M.; Liu, C.; Faiz, M.A.; Khan, M.I.; Li, T.; Cui, S. Application of particle swarm optimization and extreme learning machine forecasting models for regional groundwater depth using nonlinear prediction models as preprocessor. *J. Hydrol. Eng.* **2018**, *23*, 04018052. [CrossRef]
63. Brockwell, P.; Davies, R. *Introduction to Time Series*; Springer: New York, NY, USA, 2002.
64. Brownlee, J. Crash Course on Multi-Layer Perceptron Neural Networks. 2022. Available online: https://machinelearningmastery.com/neural-networks-crash-course/ (accessed on 7 January 2024).
65. Anderson, T.W.; Darling, D.A. A Test of Goodness-of-Fit. *J. Am. Stat. Assoc.* **1954**, *49*, 765–769. [CrossRef]
66. LibreTexts Statistics. 16.4. Box-Cox Transformations. Available online: https://stats.libretexts.org/Bookshelves/Introductory_Statistics/Introductory_Statistics_(Lane)/16%3A_Transformations/16.04%3A_Box-Cox_Transformations (accessed on 7 January 2024).
67. Yan, S.; Liu, W.; Li, X.; Yang, P.; Wu, F.; Yan, Z. Comparative Study and Improvement Analysis of Sparrow Search Algorithm. *Wirel. Comm. Mobile Comput.* **2022**, *2022*, 4882521. [CrossRef]
68. Svetunkov, I. Rolling Origin. 2003. Available online: https://cran.r-project.org/web/packages/greybox/vignettes/ro.html#:~:text=Rolling%20origin%20is%20an%20evaluation,of%20how%20the%20models%20perform (accessed on 7 January 2024).
69. AL-Allaf, O.N.A. Improving the Performance of Backpropagation Neural Network Algorithm for Image Compression/Decompression System. *J. Comp. Sci.* **2010**, *6*, 1347–1354.
70. He, K.; Sun, J. Convolutional neural networks at constrained time cost. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5353–5360.
71. Chellapilla, K.; Puri, S.; Simard, P. High performance convolutional neural networks for document processing. In Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 1 October 2006. Available online: https://inria.hal.science/inria-00112631/document (accessed on 6 January 2024).
72. 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]
73. Xu, L.; Zhang, Z.; Yao, Y.; Yu, Z. Improved Particle Swarm Optimization-Based BP Neural Networks for Aero-Optical Imaging Deviation Prediction. *IEEE Access* **2022**, *10*, 26769–26777. [CrossRef]
74. 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).
75. 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]

76. Fashae, O.; Olusola, A.; Ndubuisi, I.; Udomboso, C. Comparing ANN and ARIMA model in predictingthe discharge of River Opeki from 2010 to 2020. *River. Res. Appl.* **2019**, *35*, 169–177. [CrossRef]
77. Musarat, M.A.; Alaloul, W.S.; Rabbani, M.B.; 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. [CrossRef]
78. Senthil Kumar, A.; Sudheer, K.; Jain, S.; Agarwal, P. Rainfall-runoff modelling using artificial neural networks: Comparison of network types. *Hydrol. Process. Int. J.* **2005**, *19*, 1277–1291. [CrossRef]
79. Lilhore, U.K.; Dalal, S.; Faujdar, N.; Margala, M.; Chakrabarti, P.; Chakrabarti, T.; Simaiya, S.; Kumar, P.; Thangaraju, P.; Velmurugan, H. Hybrid CNN-LSTM model with efficient hyperparameter tuning for prediction of Parkinson's disease. *Sci. Rep.* **2023**, *13*, 14605. [CrossRef] [PubMed]
80. Ehteram, M.; Ahmed, A.N.; Khozani, Z.H.; El-Shafie, A. Graph convolutional network—Long short term memory neural network-multi layer perceptron—Gaussian progress regression model: A new deep learning model for predicting ozone concentration. *Atmos. Poll. Res.* **2023**, *14*, 101766. [CrossRef]
81. Wibawa, A.P.; Utama, A.B.P.; Elmunsyah, H.; Pujianto, U.; Dwiyanto, F.A.; Hernandez, L. Time-series analysis with smoothed Convolutional Neural Network. *J. Big. Data* **2022**, *9*, 44. [CrossRef] [PubMed]
82. 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.
83. 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.