water

Article

Daily Runoff Prediction with a Seasonal Decomposition-Based Deep GRU Method

Feifei He [1,2,3], Qinjuan Wan [4,*], Yongqiang Wang [1,2,3], Jiang Wu [1,2,3], Xiaoqi Zhang [1,2,3] and Yu Feng [1,2,3]

1 Changjiang River Scientific Research Institute, Changjiang Water Resources Commission of the Ministry of Water Resources of China, Wuhan 430010, China; hefeifei20@foxmail.com (F.H.); wangyongqiang1022@163.com (Y.W.); cky_wujiang@163.com (J.W.); zhangxq@mail.crsri.cn (X.Z.); ferrychip@foxmail.com (Y.F.)
2 Hubei Key Laboratory of Intelligent Yangtze and Hydroelectric Science, China Yangtze Power Co., Ltd., Yichang 443000, China
3 Research Center on the Yangtze River Economic Belt Protection and Development Strategy, Changjiang Water Resources Commission, Wuhan 430010, China
4 School of Economics and Business Administration, Central China Normal University, Wuhan 430010, China
* Correspondence: wanqj@ccnu.edu.cn

Abstract: Accurately predicting hydrological runoff is crucial for water resource allocation and power station scheduling. However, there is no perfect model that can accurately predict future runoff. In this paper, a daily runoff prediction method with a seasonal decomposition-based-deep gated-recurrent-unit (GRU) method (SD-GRU) is proposed. The raw data is preprocessed and then decomposed into trend, seasonal, and residual components using the seasonal decomposition algorithm. The deep GRU model is then used to predict each subcomponent, which is then integrated into the final prediction results. In particular, the hyperparameter optimization algorithm of tree-structured parzen estimators (TPE) is used to optimize the model. Moreover, this paper introduces the single machine learning model (including multiple linear regression (MLR), back propagation (BP), long short-term memory neural network (LSTM) and gate recurrent unit (GRU)) and a combination model (including seasonal decomposition–back propagation (SD-BP), seasonal decomposition–multiple linear regression (SD-MLR), along with seasonal decomposition–long-and-short-term-memory neural network (SD-LSTM), which are used as comparison models to verify the excellent prediction performance of the proposed model. Finally, a case study of the Qingjiang Shuibuya test set, which considers the period 1 January 2019 to 31 December 2019, is conducted. Case studies of the Qingjiang River show the proposed model outperformed the other models in prediction performance. The model achieved a mean absolute error (MAE) index of 38.5, a Nash-Sutcliffe efficiency (NSE) index of 0.93, and a coefficient of determination (R^2) index of 0.7. In addition, compared to the comparison model, the NSE index of the proposed model increased by 19.2%, 19.2%, 16.3%, 16.3%, 2.2%, 2.2%, and 1.1%, when compared to BP, MLR, LSTM, GRU, SD-BP, SD-MLR, SD-LSTM, and SD-GRU, respectively. This research can provide an essential reference for the study of daily runoff prediction models.

Keywords: runoff prediction; seasonal decomposition; machine learning; gated recurrent unit; hyperparameter optimization

check for updates

Citation: He, F.; Wan, Q.; Wang, Y.; Wu, J.; Zhang, X.; Feng, Y. Daily Runoff Prediction with a Seasonal Decomposition-Based Deep GRU Method. *Water* **2024**, *16*, 618. https://doi.org/10.3390/w16040618

Academic Editor: Maria Mimikou

Received: 13 December 2023
Revised: 26 January 2024
Accepted: 17 February 2024
Published: 19 February 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

Hydrological runoff forecasting is an important area of research that aims to accurately predict runoff, and can greatly improve the effectiveness of integrated watershed management for relevant departments. In the last few decades, numerous researchers have made significant progress in runoff prediction. Hydrological runoff forecasting can be divided into two categories: traditional physical models (such as the SHE model [1], Xin'anjiang model [2,3], and SWAT model [4] and data-driven models such as the autoregressive moving average model (ARMA) [5–7], multiple linear regression (MLR) [8,9], artificial neural networks (ANN) [10,11], and others [12,13].

There has been significant progress in using AI algorithms for hydrological forecasting in recent years. For instance, He [14] proposed a model that uses Bayesian model averaging with multiple machine learning models to predict medium-term streamflow. This model was tested on the Three Gorges Reservoir and demonstrated good prediction performance, proving its accuracy. Similarly, Niu [15] proposed a new streamflow prediction model with ELM-QPSO that combines the quantum particle swarm optimization (QPSO) and limit learning machine (ELM). In addition, Dai [16] proposed improving the accuracy of hydrological prediction by using an LSTM-seq2seq model for short-term water level prediction. Furthermore, Wu [17] used data-driven models, such as the long-short term memory (LSTM), gated recurrent unit models (GRU), support vector machine (SVM), and multi-layer perceptron (MLP), to predict GWL. The dynamic prediction case study of GWL in the Hebei Plain shows that the GRU model performs the best.

Forecasting reservoir inflow accurately is increasingly challenging due to the unpredictable nature of factors such as climate change and runoff. To overcome this, a combination forecasting method has been developed, which merges signal processing methods with deep learning techniques. For example, Qi [18] proposed a decomposition-ensemble learning model based on the long short-term memory neural network (DEL-LSTM) for daily reservoir inflow forecasting. The proposed DEL-LSTM model outperforms other representative models in terms of prediction accuracy. Yousefi [19] proposed a framework called causal multivariate empirical mode decomposition (CED), using it as a pre-processing step to complement a day-ahead inflow forecasting problem. Li [20] developed a hybrid model of adaptive variational mode decomposition (VMD) and bidirectional–long- and-short-term memory (Bi-LSTM), which was based on energy entropy for daily inflow forecast. The application of the models shows that the proposed model is superior to the other contrast models.

The data-driven prediction model has produced numerous research results in related fields. Grid load forecasting, similar to runoff prediction, can be studied by using the data-driven model. The field of power grid load forecasting has produced extensive research results on the deterministic and uncertain interval forecasting of power grid loads. For deterministic forecasting, He [21] proposed a hybrid model for short-term load forecasting based on variational mode decomposition (VMD) and long-short-term-memory network (LSTM). Wood [22] utilized long-short-term memory recurrent networks, applying empirical mode decomposition for feature engineering, and k-means clustering for outlier detection in load forecasting. Wang [23] developed an online algorithm called singular value thresholding (SVT), which efficiently recovers missing information by utilizing the approximate low-rank property of load data matrices. Additionally, a combinatorial deep learning method was developed that employs a multi-layer perception (MLP) neural network and a long short-term memory (LSTM) neural network with gated recurrent unit (GRU) to handle short-term and ultra-short-term load forecasting, respectively. Numerical experiments conducted on real-world load data from North China confirm the effectiveness of the proposed methodology. In addressing uncertainty forecast, He [24] proposed a new method for short-term load probability density forecasting, which can forecast the load curve for the next 24 h. This hybrid method uses a decomposition-based quantile regression forest. The simulation results have shown that the model's prediction results are well-matched with the actual load curve. Moreover, the VMD method used in this hybrid method has significant advantages in time series decomposition. More research has shown that using decomposition algorithms, neural networks, and other methods for data-driven model research is effective [25–30]. Similarly, in the field of wind and solar power prediction, numerous research has been based on data-driven methods. For instance, More [31] proposed a wind speed forecasting algorithm that uses deep learning (DL), and specifically recurrent neural networks (RNNs). A case study was conducted on time series data from windmills in the northeast of the U.S, and the research results indicated that the proposed DL-based forecasting algorithm significantly improved short-term forecasts, compared to widely used benchmark models. Sun [32] proposed a specialized CNN

"SUNSET" to predict 15-min ahead PV output. And the case study indicates that the method has a good prediction effect. Numerous related studies have shown the effectiveness of combining data-driven models with neural networks and other advanced methods, and provide good reference values for the research ideas proposed in this paper.

The main contributions of this paper include: (1) This article proposed a daily runoff prediction method with a seasonal decomposition-based deep gated recurrent unit (GRU) method (SD-GRU). The proposed hybrid method is based on decomposition and can be applied to various types of runoff time series data. (2) The model first preprocesses the raw data and then uses the seasonal decomposition algorithm to decompose the runoff series into trend, seasonal, and residual components. The deep GRU model is then used to predict each sub-component, which is then integrated into the final prediction results. (3) The tree-structured parzen estimators (TPE) are used to optimize the model. (4) This paper introduces the single machine learning model, including back propagation (BP), multiple linear regression (MLR), long-short-term memory neural network (LSTM) and Gate Recurrent Unit (GRU), as well as a combination model that includes seasonal decomposition-back propagation (SD-BP), seasonal decomposition-multiple linear regression (SD-MLR), and seasonal decomposition-long-and-short-term-memory neural network (SD-LSTM) as the comparison models that are used to verify the excellent prediction performance of the proposed model.

The remaining part of this article is structured as follows: Section 2 introduces the research methods used in this paper, and also includes the whole process of the proposed model and the model performance evaluations; Section 3 selects Qingjiang Shuibuya as the research area, and compares and analyzes the prediction performance of the proposed model and the comparative model; Section 4 uses a case study to discuss the proposed method; and Section 5 provides the conclusion of the research.

2. Methodology

2.1. Seasonal Decomposition (SD)

Seasonal decomposition algorithm is a method used to analyze seasonal changes in time series data. When analyzing time series with seasonal cycles, it is necessary to extract seasonal factors from the original time series and then analyze each factor separately. The most commonly used method of existing SD methods is the X-12-ARIMA method, which is the latest seasonal adjustment plan of the United States Census Bureau [33]. Therefore, this study will choose X-12-ARIMA as the SD method.

The X-12-ARIMA method decomposes time series X_t into three components: trend component T_t, seasonal component S_t, and residual component R_t.

$$Y_t = T_t + S_t + R_t \quad (1)$$

$$Y_t = T_t \times S_t \times R_t \quad (2)$$

where T_t is the trend component, S_t denotes the seasonal component, R_t represents the residual component, and Y_t is the original data.

In general, an additive decomposition model is suitable when the amplitude or trend cycle of seasonal fluctuations remains constant over time. When there are changes in the amplitude of seasonal fluctuations or the trend cycle over time, multiplication decomposition is a more appropriate method to use. Therefore, this article chooses the additive model to study hydrological forecasting.

2.2. Gated Recurrent Unit (GRU)

The GRU model is a variant of the LSTM model; both belong to the RNN, and are specialized in processing non-linear sequential data [34,35]. The GRU is a more efficient variant of the LSTM network, with a simpler structure that performs well [36]. Therefore, it is also a kind of network that is very manifold at present. In LSTM, three gate functions, respectively the input gate, forget gate, and output gate, are used to control the input value, memory value, and output value. In contrast, the GRU model has only two gates, namely the update gate and reset gate. The specific structure of the GRU model is shown in Figure 1.

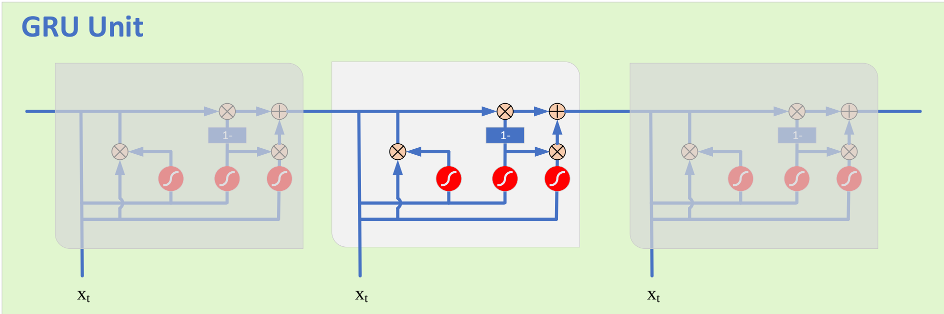

Figure 1. The structure of the GRU unit.

Usually, a neural network with three layers is capable of approximating any kind of function. however, it is easier to overfit and the results usually do not meet the requirement. Therefore, recent advances in computer power have contributed to the rapid development of deep neural networks (DNNs). The structure of DNN is shown in Figure 2.

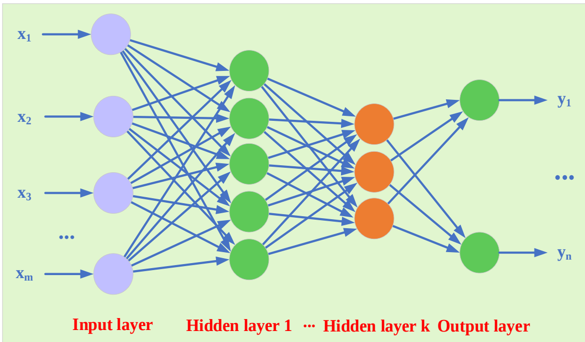

Figure 2. The structure of DNN.

2.3. Tree-Structure Parzen Estimator (TPE)

The tree-structured parzen estimators (TPE) is a global optimization Bayesian optimization algorithm based on sequence models proposed by Bergstra et al. [37]. Compared to the Gaussian process direct prediction $p(y|x)$, the TPE strategy can simultaneously obtain $p(x|y)$ and $p(y)$. One important task of hyperparameter optimization algorithms is to optimize the expected improvement (EI), which is defined as:

$$EI_{y^*}(x) := \int_{-\infty}^{\infty} \max(y^* - y, 0)\ p_M(y|x)dy \tag{3}$$

where x is the hyperparameter set, y is the measured value of the objective function at the hyperparameter set x, y^* is the threshold of the objective function, and $p(y|x)$ is a conditional probability model that represents the probability of y at hyperparameter set x.

In the tree structure Parzen estimator, the definition $p(x|y)$ is:

$$p(x|y) = \begin{cases} l(x) & if\ y < y^* \\ g(x) & if\ y > y^* \end{cases} \tag{4}$$

where y is the measured value of the objective function, and y^* is the threshold of the objective function.

When using a tree structure parzen estimator to optimize the expected increment, according to the Bayesian formula, the expected increment can be transformed into the following form:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y)p(y|x)dy = \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{p(x)}dy \tag{5}$$

Suppose $\gamma = p(y < y^*)$, $p(x) = \int_R p(x|y)p(y)dy = gl(x) + (1-g)g(x)$, the upper formula can be expressed as:

$$\int_{-\infty}^{y^*} (y^* - y)p(x|y)p(y)dy = l(x)\int_{-\infty}^{y^*} (y^* - y)p(y)dy = gy^*l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy \tag{6}$$

Therefore, the final expression for the expected increment $EI_{y^*}(x)$ is:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy}{\gamma l(x) + (1-\gamma)g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1} \tag{7}$$

From the upper equation, it can be seen that to obtain the optimal expected increment $EI_{y^*}(x)$, the hyperparameter set x should have the highest possible probability at $l(x)$ and the lowest possible probability at $g(x)$. Therefore, each iteration can obtain the maximum expected increment of the hyperparameter set x.

2.4. The Proposed Model's Entire Process

In this chapter, a daily runoff prediction method that utilizes a seasonal decomposition-based deep GRU method is proposed, which combines a seasonal decomposition algorithm with a GRU deep learning model. The proposed method's flow chart is displayed in Figure 3.

The detailed steps are as follows:

Step 1: Collate the raw data and use the augmented Dickey-Fuller test (ADF Test) method to test the stationarity of the sequence [38]. If it is not stationary, the following formula can be used to make the raw data stationary.

$$\Delta x_j = x_{i+1} - x_i, \quad (i = 0, 1, 2, \ldots N, j = 0, 1, 2, \ldots N-1) \tag{8}$$

where x_i and x_{i+1} represent the runoff data at time (i + 1) and i-th, respectively, Δx_j are runoff data at time j after the first order difference.

Then, the following formula is used to normalize the raw data.

$$X_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{9}$$

where x_i is the runoff series at the time i; x_{min}, and x_{max} represent the minimum and maximum values of the runoff sequence, respectively; and X_i is the normalized results.

Step 2: The dataset is split into three parts: the training set, the validation set, and the test set. The training set is utilized to train the model, the validation set is utilized to fine-tune the model parameters, and the test set is utilized to evaluate the predictive performance of the model. And the SD method is used to decompose the original data into trend component, seasonal component, and residual component.

Step 3: Each sub-component is trained by a deep GRU model, and rainfall data are added to each sub-component. Then, the TPE hyperparameter optimization algorithm is used to optimize the model hyperparameters, and then predict the model results.

Step 4: The final prediction results are obtained by integrating all the prediction results and denormalizing the prediction results.

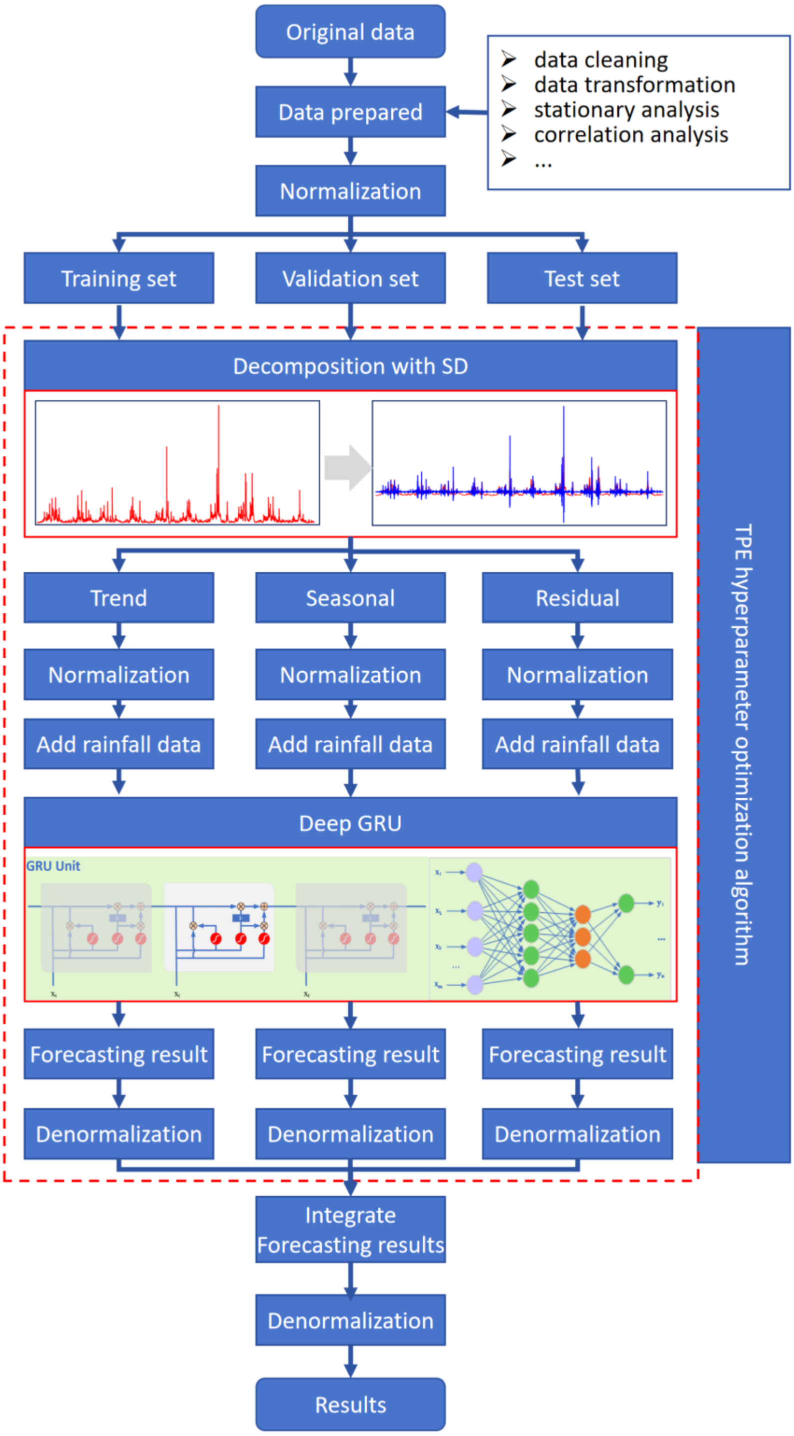

Figure 3. The flow chart of the proposed method.

2.5. Model Performance Evaluations

The accuracy evaluation of runoff prediction determines the quality of the model. Therefore, it is necessary to introduce several evaluation indexes to evaluate the prediction results comprehensively. In this paper, we introduced four evaluation indexes, including mean absolute error (MAE), root mean square error (RMSE), coefficient of determination (R^2), and Nash-Sutcliffe efficiency (NSE). The definitions of MAE, RMSE, R^2, and NSE are as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{10}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{11}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{12}$$

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{13}$$

where N denotes the number of runoff data series. y_i and $\hat{y}_i$ mean the real and prediction runoff values, respectively.

3. Case Study

3.1. Study Area Introduction

This section focuses on Qingjiang Shuibuya and illustrates how the proposed model can be effective. The Qingjiang River, located in Hubei Province, is a significant tributary of the Yangtze River in China. Its average annual flow is 464 m^3/s, and it has a drainage area of 16,700 km^2. The area receives an annual precipitation of 1415 mm. During the summer months, there is a significant amount of rainfall, with maximum runoff occurring in July and minimum runoff in January. The Qingjiang Shuibuya Power Station is the leading power station in the Qingjiang Cascade. Accurate inflow runoff prediction is critical for improving the reservoirs' overall optimization and management in the basin.

3.2. Data Analysis and Partitioning

In this section, we selected the daily runoff of Qingjiang Shuibuya in the period from 1 January 2013, to 31 December 2019 as the validation dataset for the model. Figure 4 displays the average rainfall and runoff data from the Qingjiang Shuibuya station. To train the model, we followed a flowchart and divided the dataset into three sets—training, validation, and testing. The training set was used to train the model, the validation set was used to optimize the model's hyperparameters, and the test set was used to test the model's results. As per the flowchart, we used the data from 1 January 2013, to 31 December 2017 for training; from the 1 January 2018, to 31 December 2018 for validation; and from 1 January 2019, to 31 December 2019 for testing.

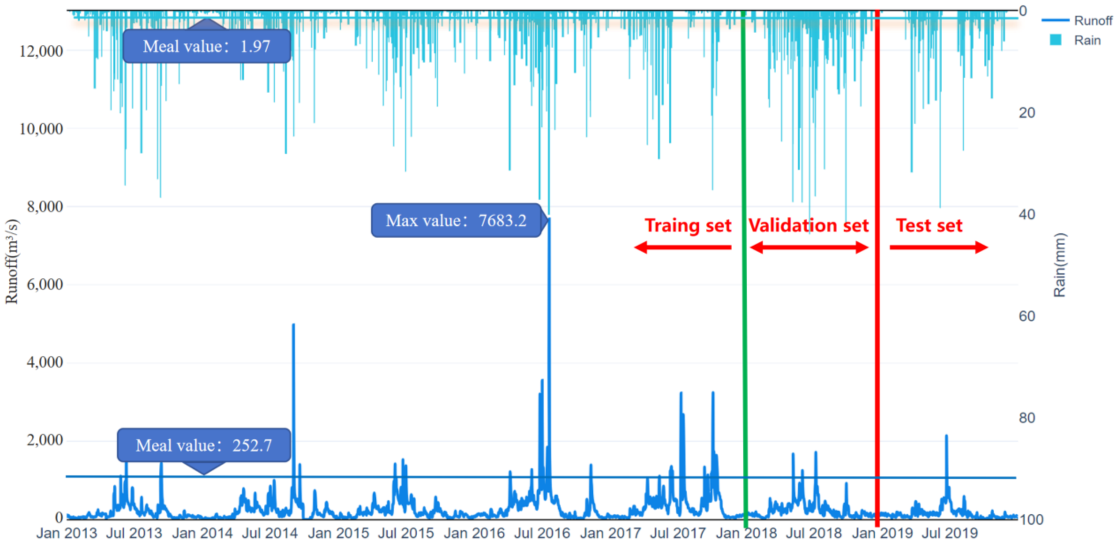

Figure 4. Average rainfall and runoff of the Qingjiang Shuibuya station.

To analyze the runoff series of Qingjiang Shuibuya, we first need to process the runoff series, including missing data compensation and data conversion. It is especially important to conduct a stability analysis of the data series, and we therefore introduce the ADF Test method to test the stability of the runoff dataset. The ADF Test method assumes that the series is not stable at first. If the test results accept the null hypothesis, the data series is not stable, and the data needs to be checked and processed to make the results stable. The results of the ADF test for the Qingjiang Shuibuya runoff series are presented in Table 1. As per the table, the test results for Qingjiang Shuibuya are less than 1%, 5%, and 10% result, and the *p*-value is close to 0. Thus, the test results reject the null hypothesis, indicating that the Qingjiang Shuibuya runoff series is stationary.

Table 1. The ADF test results for the Qingjinag Shuibuya runoff series.

Sequence	Date Range	Test Result	*p*-Value	1%	5%	10%
Runoff	1 January 2013–31 December 2019	−5.83	4.01×10^{-7}	−3.42	−2.86	−2.57

In the process of the proposed model, data processing is followed by the seasonal decomposition algorithm decomposing the runoff into three components, namely the trend component, seasonal component, and residual component. Figure 5 shows the seasonal decomposition chart of runoff.

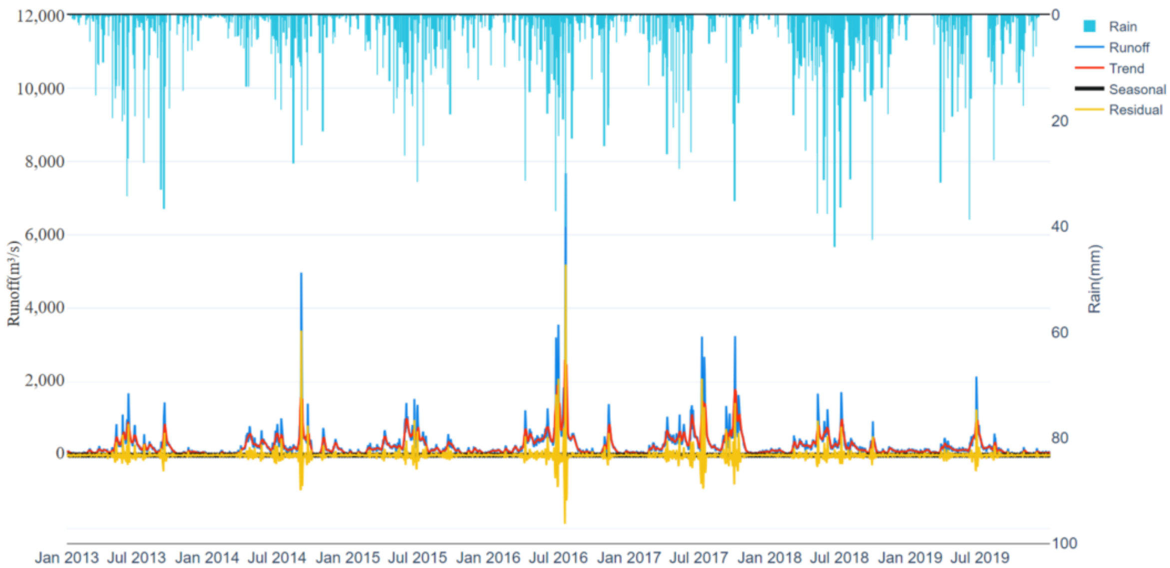

Figure 5. Seasonal decomposition chart of runoff.

3.3. Parameter Settings

In this section, we explored the parameters of various contrast methods, including both single machine learning models and combination models. We implemented these methods using Python(version: 3.9) programming languages and the TensorFlow-based Keras framework(version: 2.10.0). The single machine learning models included BP, MLR, LSTM, and GRU. For combination models, we used Seasonal Decomposition–Back Propagation (SD-BP), Seasonal Decomposition–Multiple Linear Regression (SD-MLR), Seasonal Decomposition–Long-Short-Term-Memory Neural Network (SD-LSTM), and proposed Seasonal Decomposition–Gate Recurrent Unit (SD-GRU). For the structure of the GRU model, we set one–two hidden layers of deep GRU neural network to ensure that the network can accurately fit the time series data of runoff into the reservoir and efficiently predict the runoff. We also set the loss function of the GRU model to mse, the activation function to adam, and added a dense layer to the output layer. In addition, in order to avoid overfitting, LSTM and GRU models were set to a dropout of 0.2. We also utilized partial autocorrelation function (PACF) analysis to determine the input dimension of the model. As shown in Figure 6, the lag number of the runoff series is 26, and we therefore set the input dimension of the runoff series to 26.

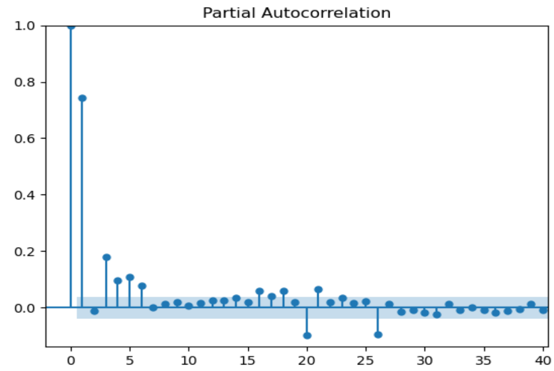

Figure 6. The autocorrelation plot of runoff data.

3.4. Results Analysis

In this section, the forecasting effects of single machine learning models, including BP, MLR, LSTM, GRU, and combination models, including SD-BP, SD-MLR, SD-LSTM, and SD-GRU were analyzed and compared. Training each model with historical runoff and rainfall data, and then using the trained models to predict daily runoff data for a total of 365 days in the period from 1 January 2019 to 31 December 2019, made it possible to analyze the predictive ability of each model. Figure 7 shows a comparison of prediction results and error release chart for different models. The scatter plots of prediction results for various models are displayed in Figure 8.

Figure 7. Comparison of prediction results and error release chart for different models.

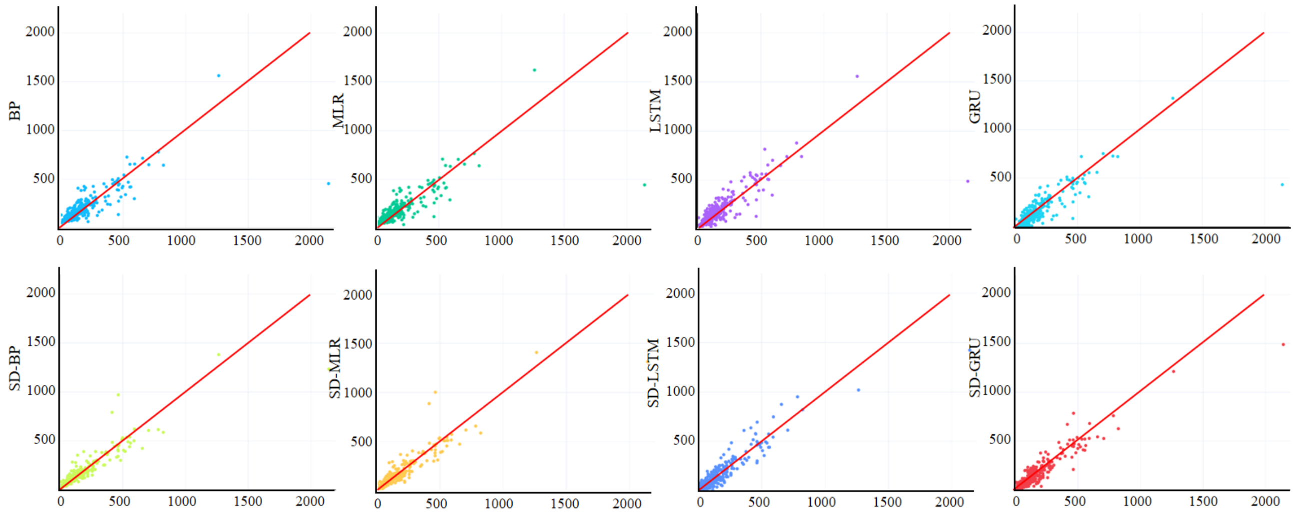

Figure 8. Scatter plots of prediction results for different models.

The bottom section of Figure 7 displays the prediction effect of the combination models, while the middle section of Figure 7 shows the prediction effect of the single machine learning models. The top part of Figure 7 shows the monthly average prediction error of different models. As shown in Figure 7, the proposed method's forecast results better fit the real values. However, predicting the peak value of daily runoff is extremely challenging, as researchers have not developed a perfect model for predicting the peak value of daily runoff. Both the proposed model and the comparative model show limitations when predicting the peak value of daily runoff. Nevertheless, the proposed model outperforms other comparative models, in terms of predicting results. Table 2 provides further details about error indicators, including MAE, MSE, NES R^2, and RMSE.

Table 2. Comparison statistics of forecast errors for each model.

Date Range	Index	BP	MLR	LSTM	GRU	SD-BP	SD-MLR	SD-LSTM	SD-GRU
1 January 2019–31 December 2019	MAE (m^3/s)	62.5	53.9	54	47.9	35.1	**34.4**	42.3	38.5
	MSE	13,301	12,634	12,227.1	12,030.4	5552.7	5364	4843.3	**4179.4**
	NSE	0.78	0.78	0.8	0.8	0.91	0.91	0.92	**0.93**
	R^2	0.59	0.61	0.62	0.63	0.83	0.83	0.85	**0.87**
	RMSE (m^3/s)	115.3	112.4	110.6	109.7	74.5	73.2	69.6	**64.6**

Note: Values in bold means the best performance of all model.

On the basis of Table 2, it can be concluded that the fitting abilities of each single machine learning model are different. The MSE, NSE, R^2, and RMSE indicators of the proposed SD-GRU model showed the best performance, with values of 4179.4, 0.93, 0.87, and 64.6, respectively. The comprehensive performance of the BP model showed the worst performance, with MAE, MSE, NSE, R^2, and RMSE indicators of 62.5, 13,301, 0.78, 0.59, and 115.3, respectively. Figure 9 shows the Taylor diagram of the proposed model and the comparison models, and Figure 10 shows the comparison chart of prediction errors of various models. Specifically, the indicators in Figure 10 have been normalized so they can be displayed in one image.

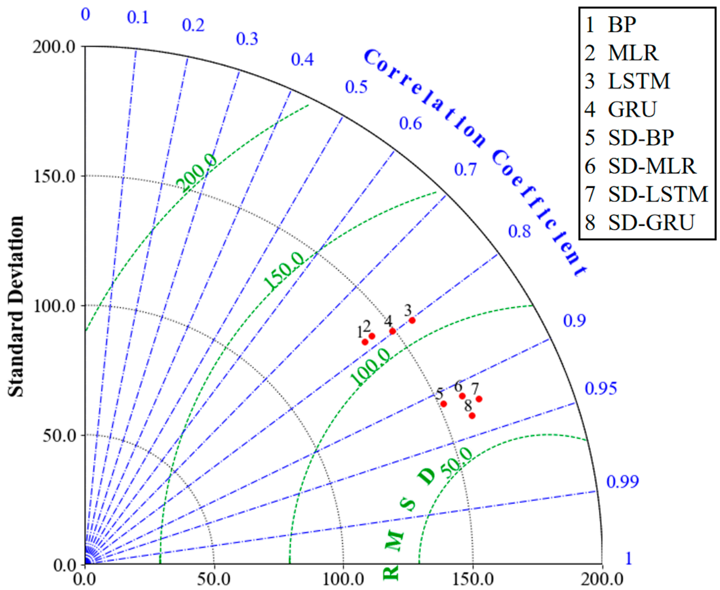

Figure 9. Taylor diagram of the proposed model and the comparison models.

Figure 10. Comparison chart showing the prediction errors of different models (all indicators have been normalized).

4. Discussion

4.1. Comparison of Prediction Performance between Single Machine Learning Models and Combination Models

In the case study section, we compared and analyzed the model using data obtained in the period 1 January 2019, to 31 December 2019 from the Qingjiang Shuibuya as the test set. Of them, we selected two major types of models, namely single machine learning models (including BP, MLR, LSTM, GRU) and combination models (including SD-BP, SD-LSTM, SD-LSTM and SD-GRU) to verify the performance of the models. We found that the combined machine learning models showed better prediction performance than the individual models. The worst performing comprehensive indicators in the combination model are SD-BP, with MAE, MSE, NSE, R^2, which showed RMSE indicators of 35.1, 5552.7, 0.91, 0.83, and 74.5, respectively. The best performing comprehensive indicators in the single machine learning model are GRU, with MAE, MSE, NSE and R^2 showing RMSE indicators of 47.9, 12030.4, 0.8, 0.63, and 109.7, respectively. It can be observed that the worst-performing combination model outperforms the best-performing single machine learning model in all indicators. On this basis, we assert that combining models can improve predictive performance.

4.2. Comparison of Prediction Performance for Different Model Parameters

In this section, we explored the impact of different parameters on the model prediction performance, and used the TPE hyperparameter optimization algorithm to optimize the parameters and minimize the MSE index of the validation set to obtain the optimal hyperparameter sets. The results showed that the model had the best prediction performance when the number of hidden layers was two, the number of nodes in the first layer was five, and the number of nodes in the second layer was 10. We selected seven models with different numbers of hidden layers and nodes (including one hidden layer with 10, 50, and 100 nodes, and two hidden layers with 5–10, 10–20, 25–50, and 50–100 nodes, respectively) for comparison. Table 3 shows a comparison chart that sets out prediction indicators for different hyperparameter models. Of them, Nodes (10) represents one hidden layer with 10 nodes, and Nodes (10–20) represents two hidden layers, with 10 nodes in the first layer and 20 nodes in the second layer. As in Figure 10, the indicators in Figure 11 have been normalized so they can be displayed in one image.

Table 3. Comparison of the prediction performance of proposed models SD-GRU that have different numbers of neural network nodes.

Date Range	Index	Nodes (10)	Nodes (50)	Nodes (100)	Nodes (5–10)	Nodes (10–20)	Nodes (25–50)	Nodes (50–100)
1 January 2019–31 December 2019	MAE (m^3/s)	66.6	57	47.7	**38.5**	40.8	48.4	44.9
	MSE	14,871	11,023.9	7699.1	**4179.4**	5178.6	6822.1	7149
	NSE	0.75	0.81	0.87	**0.93**	0.92	0.89	0.88
	R^2	0.54	0.66	0.76	**0.87**	0.84	0.79	0.78
	RMSE (m^3/s)	121.9	105	87.7	**64.6**	72	82.6	84.6

Note: Values in bold means the best performance of all model.

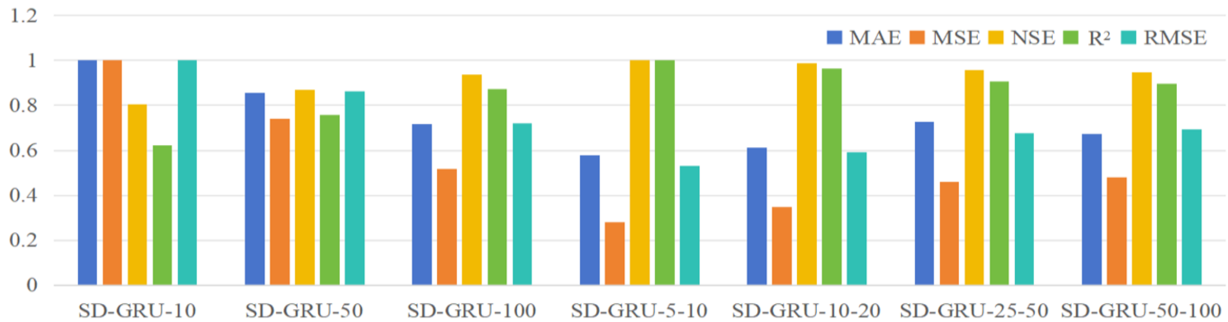

Figure 11. Comparison chart of prediction indicators for different hyperparameter models (all indicators have been normalized).

On the basis of Table 3 and Figure 11, it is clear that the model's prediction performance varies significantly in different hyperparameter sets. The model with 10 nodes in the first hidden layer and 20 nodes in the second layer shows the best prediction performance, with MAE, MSE, NSE, R^2, and RMSE indicators of 38.5, 4179.4, 0.93, 0.87, and 64.6, respectively. Additionally, models with two hidden layers perform better than those with one layer. In models with one hidden layer, the greater the number of neural network nodes, the better the prediction performance. However, in models with two hidden layers, the greater the number of neural network nodes, the worse the prediction performance. This could be due to the fact that in models with one hidden layer, a larger number of neural network nodes can better fit the runoff sequence. However, in models with two hidden layers, there may not be sufficient training samples to train the model when the number of neural network nodes increases, which could result in the model underfitting, leading to poorer forecast performance.

5. Conclusions

In this article, we propose to predict daily runoff by using a seasonal decomposition-based deep GRU method. The proposed model preprocesses the data and uses the seasonal decomposition method to decompose the runoff sequence into trend components, seasonal components, and residual components. Each component is predicted using a deep GRU model. The TPE hyperparameter optimization algorithm is then used to optimize the hyperparameters of the proposed model. To verify the performance of the proposed model, several single machine learning models, including BP, MLR, LSTM, GRU, and combination models, including SD-BP, SD-LSTM, and SD-LSTM, were chosen as comparison methods. The study offers the following conclusion:

(1) By drawing on the prediction results of the proposed SD-GRU model and comparative models of the test set conducted in the period 1 January 2019 to 31 December 2019 at the Qingjiang Shuibuaya, we conclude that the proposed model exhibits the best forecasting performance. The MAE, MSE, NSE, R^2, and RMSE indicators of the proposed model showed the best performances, with values of 38.5, 4179.4, 0.93, 0.87, and 64.6, respectively.

(2) By comparing the single machine learning models, including BP, MLR, LSTM, GRU, and combination models, including SD-BP, SD-LSTM, SD-LSTM, and SD-GRU, it was observed that the prediction performance of combination models was superior to that of single machine learning models in all indicators. On this basis, it was asserted that combining models can improve prediction performance.

(3) A comparison of different model parameters of GRU neural networks showed that, within a certain range, the greater the number of nodes of the single hidden layer model, the better the prediction effect of the model. However, in the multi-hidden layer model, the greater the number of nodes, the worse the prediction performance, which was due to the influence of the insufficient number of training samples. On the whole, it was however found that the prediction performance of multi-hidden layer neural networks was better than that of single-hidden layer models.

Author Contributions: Conceptualization, F.H.; data curation, Q.W. and Y.W.; investigation, Y.W.; methodology, F.H. and Q.W.; software, F.H. and Y.F.; validation, J.W. and X.Z.; visualization, J.W.; Writing—original draft, F.H.; writing—review & editing, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program Youth Scientist Project of China, as part of a project entitled "Medium and Long-term Water Resource Prediction Technology for the Water Source Area of the Middle Route of the South-to-North Water Transfer Project" (2023YFC3210500); the Natural Science Foundation of Hubei Province (2021CFB151, 2022CFD027); the central research institutes of Basic Research and Public Service Special Operations (CKSF2021441/SZ); the Key Project of Chinese Water Resources Ministry (SKS-2022120); the National Natural Science Foundation of China (No. 42271044, No. 52109003, No. 52009005).

Data Availability Statement: The original data cannot be made public, due to our confidentiality agreements with relevant departments.

Acknowledgments: Various Python open-source frameworks were used in this study. We would like to express our gratitude to all contributors. We would also like to give a special thanks to the anonymous reviewers and editors for their constructive comments.

Conflicts of Interest: Feifei He, Yongqiang Wang, Jiang Wu, Xiaoqi Zhang and Yu Feng were employed by China Yangtze Power Co., Ltd. The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Abbott, M.B.; Bathurst, J.C.; Cunge, J.A.; O'Connell, P.E.; Rasmussen, J. An introduction to the European Hydrological System—Systeme Hydrologique Europeen,"SHE", 2: Structure of a physically-based, distributed modelling system. *J. Hydrol.* **1986**, *87*, 61–77. [CrossRef]

2. Zhao, R.-J. The Xinanjiang model applied in China. *J. Hydrol.* **1992**, *135*, 371–381. [CrossRef]
3. Cheng, C.T.; Ou, C.P.; Chau, K.W. Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall–runoff model calibration. *J. Hydrol.* **2002**, *268*, 72–86. [CrossRef]
4. Arnold, J.G.; Srinivasan, R.; Muttiah, R.S.; Williams, J.R. Large area hydrologic modeling and assessment part I: Model development 1. *Jawra J. Am. Water Resour. Assoc.* **1998**, *34*, 73–89. [CrossRef]
5. Wang, W.; Chau, K.; Cheng, C.; Qiu, L. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *J. Hydrol.* **2009**, *374*, 294–306. [CrossRef]
6. Lin, G.; Chen, G.; Huang, P. Effective typhoon characteristics and their effects on hourly reservoir inflow forecasting. *Adv. Water Resour.* **2010**, *33*, 887–898. [CrossRef]
7. Castellano-Méndez, M.; González-Manteiga, W.; Febrero-Bande, M.; Prada-Sánchez, J.M.; Lozano-Calderón, R. Modelling of the monthly and daily behaviour of the runoff of the Xallas river using Box–Jenkins and neural networks methods. *J. Hydrol.* **2004**, *296*, 38–58. [CrossRef]
8. Bashir, A.; Shehzad, M.A.; Hussain, I.; Rehmani, M.I.A.; Bhatti, S.H. Reservoir inflow prediction by ensembling wavelet and bootstrap techniques to multiple linear regression model. *Water Resour. Manag.* **2019**, *33*, 5121–5136. [CrossRef]
9. Tsakiri, K.; Marsellos, A.; Kapetanakis, S. Artificial neural network and multiple linear regression for flood prediction in Mohawk River, New York. *Water-Sui* **2018**, *10*, 1158. [CrossRef]
10. Jain, S.K.; Das, A.; Srivastava, D.K. Application of ANN for reservoir inflow prediction and operation. *J. Water Res. Plan. Man.* **1999**, *125*, 263–271. [CrossRef]
11. Xu, Z.X.; Li, J.Y. Short-term inflow forecasting using an artificial neural network model. *Hydrol. Process.* **2002**, *16*, 2423–2439. [CrossRef]
12. Mouatadid, S.; Adamowski, J. Using extreme learning machines for short-term urban water demand forecasting. *Urban Water J.* **2017**, *14*, 630–638. [CrossRef]
13. Kisi, O. Pan evaporation modeling using least square support vector machine, multivariate adaptive regression splines and M5 model tree. *J. Hydrol.* **2015**, *528*, 312–320. [CrossRef]
14. He, F.; Zhang, H.; Wan, Q.; Chen, S.; Yang, Y. Medium Term Streamflow Prediction Based on Bayesian Model Averaging Using Multiple Machine Learning Models. *Water-Sui* **2023**, *15*, 1548. [CrossRef]
15. Niu, W.; Feng, Z.; Cheng, C.; Zhou, J. Forecasting daily runoff by extreme learning machine based on quantum-behaved particle swarm optimization. *J. Hydrol. Eng.* **2018**, *23*, 4018002. [CrossRef]
16. 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]
17. Wu, Z.; Lu, C.; Sun, Q.; Lu, W.; He, X.; Qin, T.; Yan, L.; Wu, C. Predicting Groundwater Level Based on Machine Learning: A Case Study of the Hebei Plain. *Water-Sui* **2023**, *15*, 823. [CrossRef]
18. Qi, Y.; Zhou, Z.; Yang, L.; Quan, Y.; Miao, Q. A Decomposition-Ensemble Learning Model Based on LSTM Neural Network for Daily Reservoir Inflow Forecasting. *Water Resour. Manag.* **2019**, *33*, 4123–4139. [CrossRef]
19. Yousefi, M.; Cheng, X.; Gazzea, M.; Wierling, A.H.; Rajasekharan, J.; Helseth, A.; Farahmand, H.; Arghandeh, R. Day-ahead inflow forecasting using causal empirical decomposition. *J. Hydrol.* **2022**, *613*, 128265. [CrossRef]
20. Li, F.; Ma, G.; Chen, S.; Huang, W. An ensemble modeling approach to forecast daily reservoir inflow using bidirectional long-and short-term memory (Bi-LSTM), variational mode decomposition (VMD), and energy entropy method. *Water Resour/ Manag/* **2021**, *35*, 2941–2963. [CrossRef]
21. He, F.; Zhou, J.; Feng, Z.; Liu, G.; Yang, Y. A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm. *Appl. Energ.* **2019**, *237*, 103–116. [CrossRef]
22. Wood, M.; Ogliari, E.; Nespoli, A.; Simpkins, T.; Leva, S. Day Ahead Electric Load Forecast: A Comprehensive LSTM-EMD Methodology and Several Diverse Case Studies. *Forecasting* **2023**, *5*, 297–314. [CrossRef]
23. Wang, X.; Duan, Z.; Liu, L.; Li, M.; An, Y.; Zhou, Y. Multi-Timescale Load Forecast of Large Power Customers Based on Online Data Recovery and Time Series Neural Networks. *J. Circuits Syst. Comput.* **2022**, *31*, 2250088. [CrossRef]
24. He, F.; Zhou, J.; Mo, L.; Feng, K.; Liu, G.; He, Z. Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest. *Appl. Energ.* **2020**, *262*, 114396. [CrossRef]
25. Falces, A.; Capellan-Villacian, C.; Mendoza-Villena, M.; Zorzano-Santamaria, P.J.; Lara-Santillan, P.M.; Garcia-Garrido, E.; Fernandez-Jimenez, L.A.; Zorzano-Alba, E. Short-term net load forecast in distribution networks with PV penetration behind the meter. *Energy Rep.* **2023**, *9*, 115–122. [CrossRef]
26. Sharma, M.; Mittal, N.; Mishra, A.; Gupta, A. Machine Learning-Based Electricity Load Forecast for the Agriculture Sector. *Int. J. Softw. Innov.* **2023**, *11*, 27. [CrossRef]
27. Nespoli, A.; Ogliari, E.; Pretto, S.; Gavazzeni, M.; Vigani, S.; Paccanelli, F. Electrical Load Forecast by Means of LSTM: The Impact of Data Quality. *Forecasting* **2021**, *3*, 91–101. [CrossRef]
28. Singh, P.; Dwivedi, P. Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem. *Appl. Energ.* **2018**, *217*, 537–549. [CrossRef]
29. Krishna, A.B.; Abhyankar, A.R. Normal-transformation-based probabilistic load flow with correlated wind and load forecast errors. *Renew. Energy Focus* **2022**, *43*, 117–129. [CrossRef]

30. Safari, M.J.S.; Arashloo, S.R.; Mehr, A.D. Rainfall-runoff modeling through regression in the reproducing kernel Hilbert space algorithm. *J. Hydrol.* **2020**, *587*, 125014. [CrossRef]
31. Ghaderi, A.; Sanandaji, B.M.; Ghaderi, F. Deep forecast: Deep learning-based spatio-temporal forecasting. *arXiv* **2017**, arXiv:1707.08110.
32. Sun, Y.; Venugopal, V.; Brandt, A.R. Short-term solar power forecast with deep learning: Exploring optimal input and output configuration. *Sol Energy* **2019**, *188*, 730–741. [CrossRef]
33. Staff, T.S. *X-12-ARIMA Reference Manual*; Citeseer: State College, PA, USA, 2002.
34. Williams, R.J.; Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1989**, *1*, 270–280. [CrossRef]
35. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
36. Pezeshki, M. *Sequence Modeling Using Gated Recurrent Neural Networks*; Cornell University: Ithaca, NY, USA, 2015.
37. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 1–9.
38. Mushtaq, R. Augmented Dickey Fuller Test. 2011. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1911068 (accessed on 11 February 2024).

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.