water

MDPI

Article

Computing River Discharge Using Water Surface Elevation Based on Deep Learning Networks

Wei Liu [1], Peng Zou [1,*], Dingguo Jiang [2], Xiufeng Quan [3] and Huichao Dai [2]

[1] College of Hydraulic and Environmental Engineering, China Three Gorges University, Yichang 443002, China; lw791231@aliyun.com
[2] China Three Gorges Corporation, Wuhan 430010, China; jiangdingguo@aliyun.com (D.J.); dai_huichao@263.net (H.D.)
[3] Key Laboratory of Coastal Disaster and Defense, Ministry of Education, Hohai University, Nanjing 210098, China; quanxiufeng@hhu.edu.cn
* Correspondence: cruisezou@163.com

Abstract: Accurately computing river discharge is crucial, but traditional computing methods are complex and need the assistance of many other hydraulic parameters. Therefore, it is of practical value to develop a convenient and effective auto-computation technique for river discharge. Water surface elevation is relatively easy to obtain and there is a strong relationship between river discharge and water surface elevation, which can be used to compute river discharge. Unlike previous usage of deep learning to predict short-term river discharge that need multiple parameters besides water level, this paper proved that deep learning has the potential to accurately compute long-term river discharge purely based on water level. It showed that the majority of relative errors on the test dataset were within ±5%, particularly it could operate continuously for almost one year with high precision without retraining. Then, we used BiGRU to compute river flow with different hyperparameters, and its best RMSE, NSE, MAE, and MAPE values were 256 m^3/s, 0.9973, 207 m^3/s, and 0.0336, respectively. With this data-driven based technology, it will be more convenient to obtain river discharge time series directly from local water surface elevation time series accurately in natural rivers, which is of practical value to water resources management and flood protection.

Keywords: river flow; water level; river stage; deep learning networks; RNN; Yangtze River

Citation: Liu, W.; Zou, P.; Jiang, D.; Quan, X.; Dai, H. Computing River Discharge Using Water Surface Elevation Based on Deep Learning Networks. *Water* **2023**, *15*, 3759. https://doi.org/10.3390/w15213759

Received: 5 October 2023
Revised: 25 October 2023
Accepted: 25 October 2023
Published: 27 October 2023

1. Introduction

RD (river discharge) is a critical hydrologic variable that links atmospheric, oceanic, and terrestrial processes, which plays a key role in addressing various aspects such as assessing food risks and guiding hydropower stations' operation. The Global Climate Observing System (GCOS) considers it essential to understanding the hydrological cycle and managing water supplies [1]. In most cases, however, RD cannot be measured directly and needs relevant hydraulic parameters to compute its value, such as flow rate at multiple points and cross-section area. Thus, it is meaningful to develop a convenient RD computation method purely based on water level.

The watershed model assumes a pivotal role in the calculation of river discharge using conventional hydrodynamic principles. This approach offers the capacity to encompass a designated regional area and furnish precise data on water flow characteristics, alongside key water quality parameters, such as EFDC (Environmental Fluid Dynamics Code) [2–4]. Nevertheless, these numerical simulation methods necessitate a lot of effort in grid generation, and the computational process is time consuming.

Nowadays, monitoring discharge in one river cross is replaced by many indirect methods, such as SDRC (stage-discharge rating curve) [5], IVRC (index velocity rating curve) [6,7], and CSA (continuous slope-area) [8–10]. These approaches estimate RDs by collecting some important hydraulic parameters, and they are widely used around

the world. SDRC observes the local WSE (water surface elevation) and converts it into RD through the rating curve. SDRC's limitation is obvious, because it cannot recognize two different hydraulic phases, including rising and falling, and the hysteresis will lead to significant bias [11]. Consequently, IVRC takes the mean velocity of river cross-section into consideration, and hopes to improve RD computation's accuracy [6]. However, in practical terms, this improvement still cannot achieve satisfying performances [10]. For CSA, it is a relatively novel method, which uses two WSE observations and calculates the RD based on the Manning equation, and this method has been used in some small rivers with mild slopes [10].

It can be seen that WSE plays an important role in computing RD, and the key component of RD computation is the relationship between the WSE and RD. However, there are many limitations affecting accurate RD computation, for instance, non-stationarities of river flow [12], the variety of hydraulic roughness [13], and human activities [14]. Considering these factors, scholars worldwide have designed many correction ways, including the use of physical parameters in rating curve formulations [15] and hydraulic models [16,17].

Despite the various correction approaches provided, these correction protocols are based on purely empirical or semi-empirical approaches or analytical models and will bring additional operational costs [10]. Moreover, existing correction methods for rating curves have typically neglected the temporal continuity in both of RD and WSE. Since traditional methods estimate RD using WSE at discrete time points, these methods are unable to recognize different hydraulic processes (rising or falling). In reality, RD and WSE are typical time series data, with each value tightly connected to neighboring values. Therefore, using WSE sequences to estimate corresponding RD sequences may be more effective than the single point to single point computation from rating curves. A continuous WSE time series will most likely help us distinguish whether the river stage is rising or not.

Some deep learning networks are born to handle data with time steps, RNN (recurrent neural network) for instance, their hidden states can be passed along time steps, and RNNs (RNN and its variants) are successfully employed to various kinds of hydrology tasks [18–25]. In particular, to deal with short-term runoff prediction problems, a deep learning multi-dimensional ensemble method has proven to be effective [26,27]. LSTM (long short-term memory) is also used to improve runoff forecasting performance through error predictions [28], which has shown excellent results in the Russian River basin, California, the United States. For medium to long-term water level prediction tasks, an improved spatio-temporal attention mechanism has been designed for LSTM [29]. Additionally, LSTM is combined with a Seq2Seq (sequence-to-sequence) learning structure to predict the errors in the forecasted runoff from a hydrological model [30].

In this study, we used deep learning networks to compute RD time series accurately based on WSE time series only, rather than approximately forecasting, which is different from previous applications. Compared to conventional computing approaches, this data-driven based technique is more convenient and accurate. In detail, we evaluate the performance of eight different deep learning networks using the dataset observed from Zhutuo gauging station on the Yangtze River. Then, the BiGRU (bi-directional gated recurrent unit) network is selected as the representative model to explore the effects of different hyperparameters on the outcomes. Finally, we test the applicability of the BiGRU by applying it to six gauging stations along the Yangtze River, including both tributaries and the mainstream.

2. Materials and Methods

2.1. Study Area and Data

2.1.1. Study Area

In this paper, the Yangtze River is selected to be the research objective, because of its various, complex and diverse hydraulic properties along the different river cross sections, which will be helpful to adequately examine the DLNs' RD computation effects.

It is well-known that the Yangtze River basin boasts a subtropical monsoon climate characterized by frequent heavy rainfall and abundant water resources. The Yangtze River basin exhibits a significant variation in yearly average precipitation, spanning from 300 to 2400 mm from west to east, accompanied by a considerable annual average temperature range from 9 to 18 °C [31]. Besides, the Yangtze River in China is the largest river in Asia and the third largest around the world, originating from Tangela Mountain in Qinghai Province and running approximately 6300 km through over 7000 tributaries. The upper reaches of the Yangtze River, stretching 4504 km with a basin area of 106 km^2, are located above Yichang and hold a strategic position in China's water resource management and regulation. Figure 1 shows the location of the Yangtze River basin and the hydrologic observation stations used in this paper.

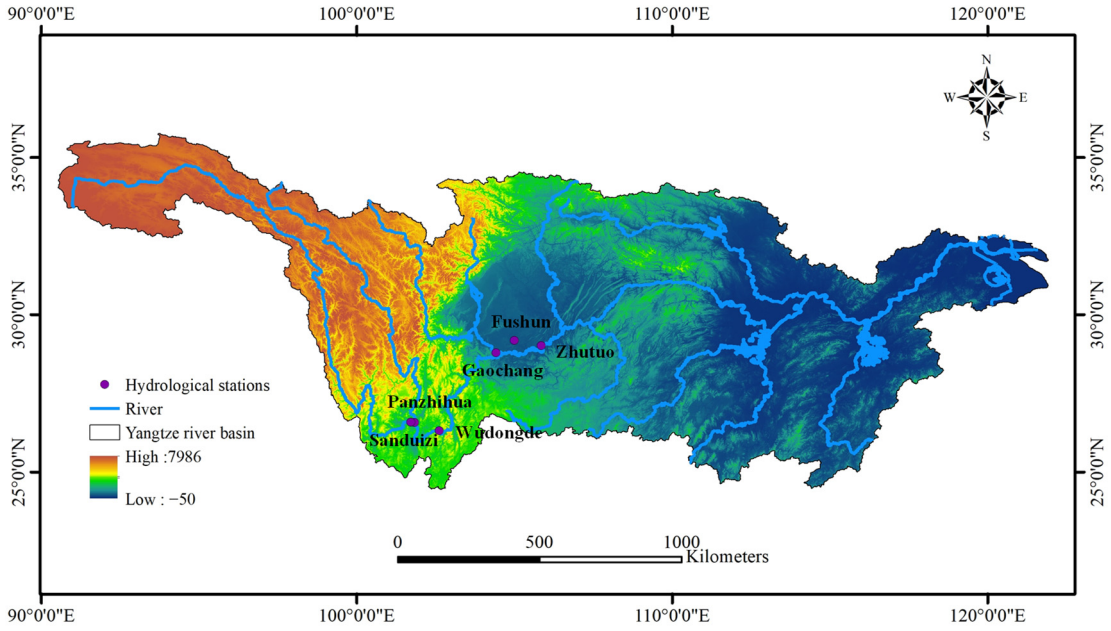

Figure 1. Location of the Yangtze River basin and hydrologic observation stations located within the basin.

2.1.2. Data Description

In this study, the possibility of using water level data to directly compute river flow is explored. To ensure the training effect, the data collected in this paper is re-organized and verified by experts in every station according to water quantity balance throughout a year, which means it will reduce the errors introduced by conventional computation methods. The data used in this study consists of water level and river flow data collected from six different stations with various hydrodynamic conditions in the Yangtze River basin. Each location has a training dataset that covers one year, including 2020 for Gaochang and Fushun, and 2018 for Panzhihua, Sanduizi, Wudongde, and Zhutuo. The corresponding test dataset covers the subsequent year, including 2021 for Gaochang and Fushun, 2019 for Panzhihua, Sanduizi, Wudongde, and August 2015 to March 2016 for Zhutuo. The relationships between river stage and river discharge in different stations are shown in Figure 2. The river discharge distribution in different stations can be seen in Figure 3. The river discharge in Gaochang has the largest change and Panzhihua has the smallest change. Zhutuo has the largest average river discharge and Fushun has the smallest.

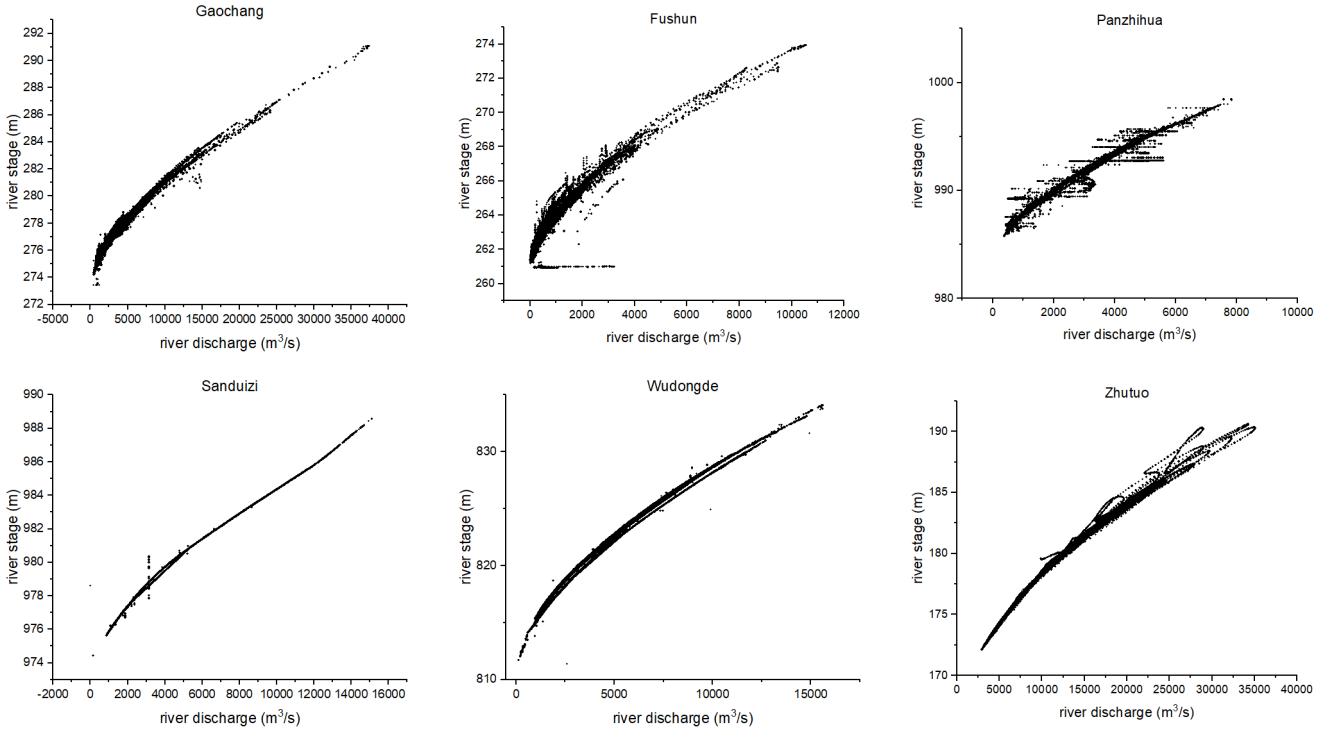

Figure 2. The relationships between river stage and river discharge at different stations.

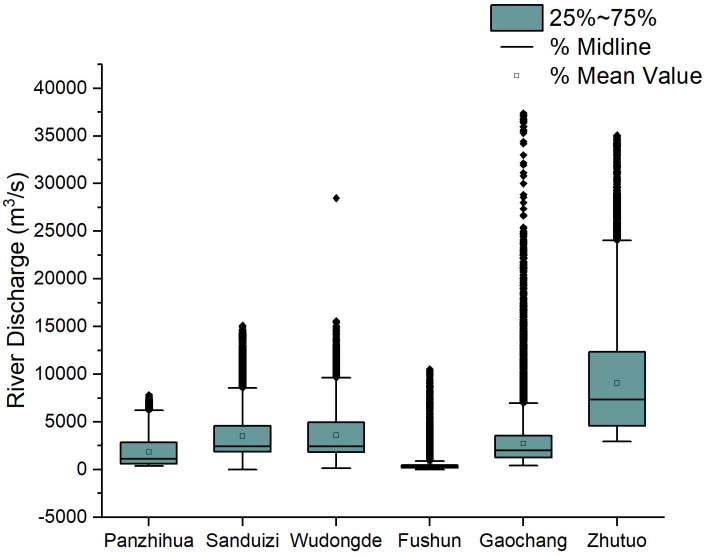

Figure 3. River discharge distribution at different stations.

Water level is used as the input variable in the deep learning networks, while river flow is the output variable. Prior to analysis, the data was preprocessed, including cleaning and filtering for outliers, and scaling of the input and output variables to a common range. To compare the performance of different deep learning networks and hyperparameters, we use the data from Zhutuo as a reference to evaluate their performance under the same conditions. Then, we select one of the appropriate networks and a combination of hyperparameters, and apply it to the data from all six stations and observe the network's results in various sites.

2.2. Deep Learning Networks

There are eight kinds of DLNs employed in this paper, including GRU (gated recurrent unit), BiGRU (bi-directional gated recurrent unit), LSTM (long short-term memory), BiLSTM (bi-directional long short term memory), RNN (recurrent neural network), BiRNN (bi-directional recurrent neural network), Seq2Seq (sequence-to-sequence), and Seq2Seq attention (sequence-to-sequence with attention mechanism).

RNN's hidden states can be passed along time steps, and every time step's output is calculated by the previous hidden state and the current input, as shown in Figure 4a. Therefore, RNN's unique structure has a huge advantage in coupling with time series. However, when RNN gets deeper, the gradient explosion will significantly affect RNN's performance, so LSTM and GRU [32] are proposed to solve this problem. LSTM brings an output gate, input gate and forget gate in hidden state, and the basic unit of LSTM is displayed in Figure 4b. GRU adopts the reset gate and update gate in the hidden state, and the basic unit of GRU is displayed in Figure 4c. As GRU has less parameters in hidden state than LSTM, but achieves the same function, GRU usually has a faster calculation speed than LSTM, and the performances are very close [33]. Bi-directional DLNs can use information from past and future observations to estimate current observations because they add a hidden layer that passes information in reverse, as shown in Figure 4d. Seq2Seq [32,34] uses a structure of encoder and decoder to solve sequence to sequence problems. The encoder copes with input sequences and transforms them into context, and the decoder then concatenates this context and the input of the decoder to predict the output sequences, as shown in Figure 4e. As not every piece of context is associated with the output sequence, the Bahdanau attention mechanism comes up [35], which allows the decoder to predict output with relevant context, as shown in Figure 4f.

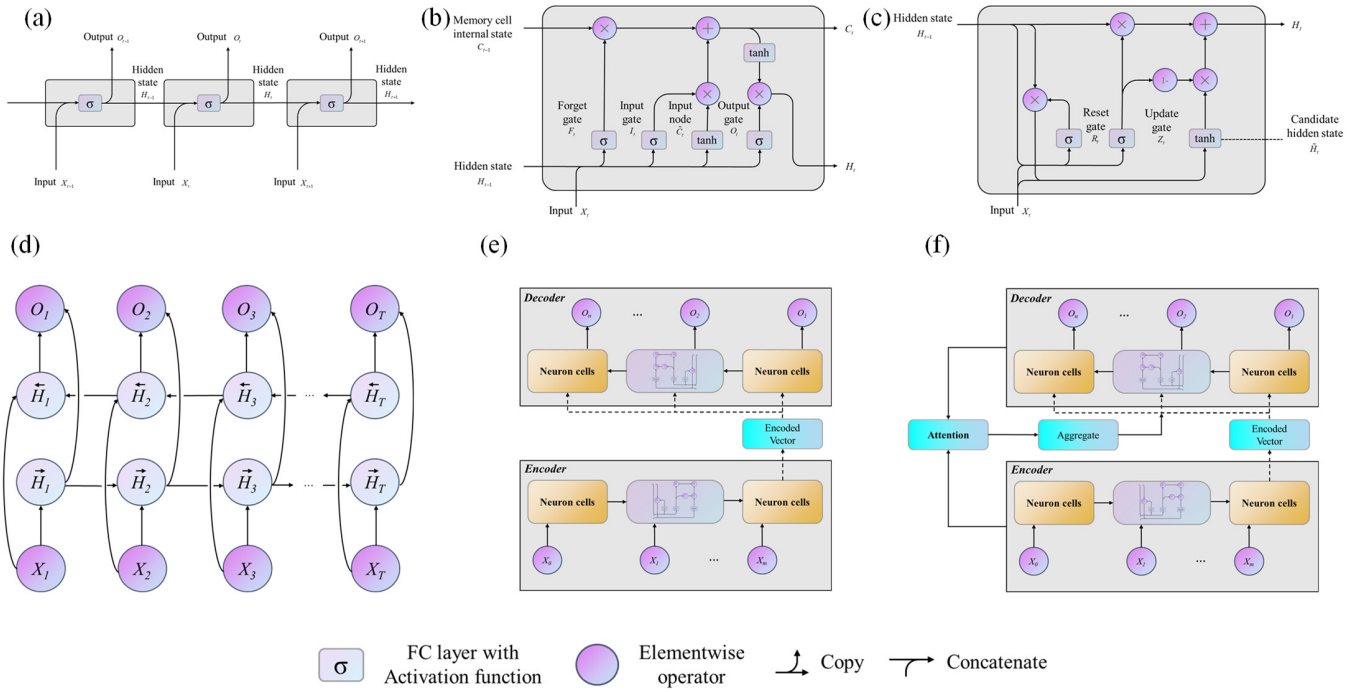

Figure 4. Basic structure of deep learning networks. (**a**) Workflow of RNN, (**b**) the basic unit of LSTM, (**c**) the basic unit of GRU, (**d**) the typical structure of Bi-direction networks, (**e**) the typical structure of Seq2eq, and (**f**) the typical structure of Seq2eq attention.

To further elaborate the workflow in RNNs, Figure 5a displays the data structure and Figure 5b shows the hidden state passing along the time steps. There are three dimensions in the data, which contain batch size, input size, and time steps. Input sequences are sent

to the input layers, where the bias between the output sequences and the target sequences will be calculated, which will then be used to update the parameters in the DLNs.

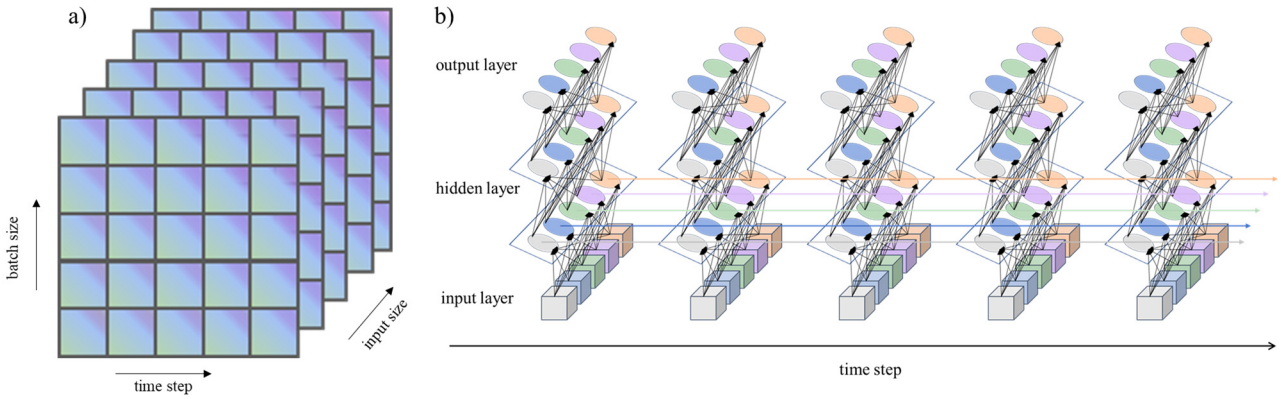

Figure 5. (**a**) The data structure; (**b**) the hidden state passing along the time steps.

2.3. Computation Procedure Design

2.3.1. Computation Procedure

Deep learning networks are leveraged to precisely compute RD time series solely from WSE time series, departing from the conventional practice of approximate forecasting found in prior applications. In comparison to traditional computation methods, this data-driven approach offers enhanced convenience while maintaining high levels of accuracy. To estimate river flow from water level directly using deep learning networks, the following five steps are recommended:

(1) Collect measured data: Collect data of river flow and water level and ensure they cover the same time period. Divide the dataset into training and testing sets and perform any necessary pre-processing. (2) Cut the sequences: Divide both the river flow and water level sequences into shorter series of equal length (t). Shuffle the series to ensure diversity in training, but ensure that each water level series corresponds to the corresponding river flow series. (3) Train the network: Send n (batch size) water level series into the deep learning network at one time. Compare the network output with the corresponding river flow and compute the loss. Use this loss to update the network parameters and repeat this process until the network converges. (4) Test the network: After training the network, input new series of water level into the network and obtain the estimated river flow. (5) Evaluate the performance: Evaluate the performance of the deep learning network by comparing the estimated river flow with the actual measured river flow, using appropriate metrics such as the mean squared error, coefficient of determination, or other relevant measures.

Besides, all deep learning networks in this study were implemented using the Pytorch 1.12.1 framework in Python 3.8, with GPU acceleration provided by CUDA 11.3.1. Each network was trained for a duration of 24 h on a computing system comprising Intel i7-11700K CPU, Nvidia RTX 2060 GPU, and 32 GB of RAM operating at 3200 MHz.

2.3.2. Evaluation Metrics

The four metrics used in this study to evaluate the performance of deep learning networks are relative mean absolute percentage error (RMSE), Nash–Sutcliffe efficiency coefficient (NSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). This research chose these metrics because they are widely used in previous literatures [36–39]. Smaller values of MAE, RMSE, and MAPE indicate better performance of the network in terms of the difference between measured and estimated data. On the other hand, NSE quantifies the closeness between the estimated and measured data, and a value of NSE closest to 1 indicates the highest computation accuracy. The expressions for these metrics are given below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_p - y_o)^2} \quad (1)$$

$$NSE = 1 - \frac{\sum_{i=1}^{n}(y_o - y_p)^2}{\sum_{i=1}^{n}(y_o - \overline{y}_o)^2} \quad (2)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_p - y_o| \quad (3)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_o - y_p}{y_o}\right| \quad (4)$$

where n is the number of inputs, y_o is the observed data, y_p is the estimated, and $\overline{y}_o$ is the average value of the observed data.

3. Results

3.1. Comparison of the Effect of Different Deep Learning Networks

Based on the measured river flow and water level in Zhutuo station, which is located in the middle of Yangtze River, this study employed eight deep learning networks to estimate river flow directly from the water level. The performance of each network was evaluated on both the training dataset and the test dataset, as shown in Figure 6.

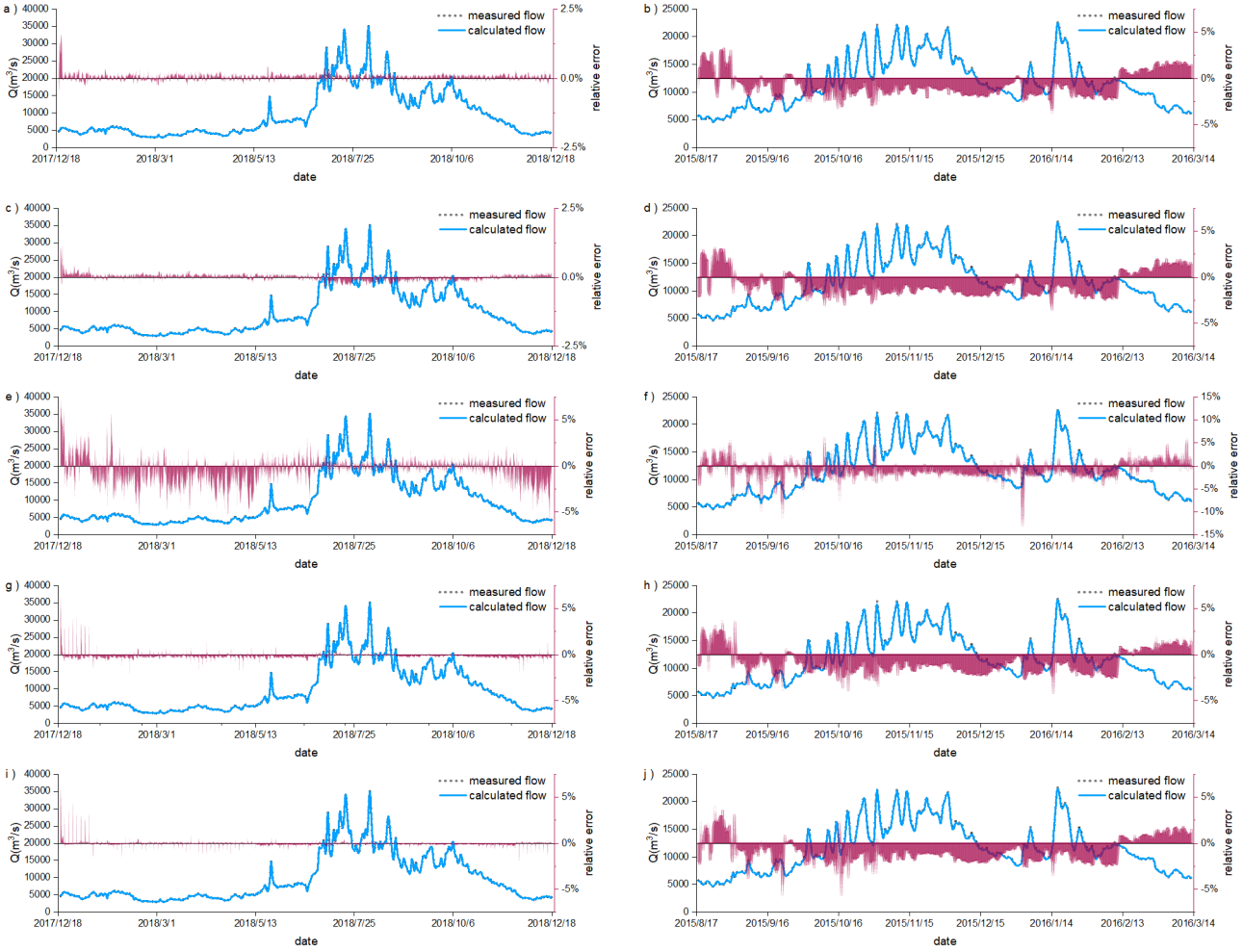

Figure 6. *Cont.*

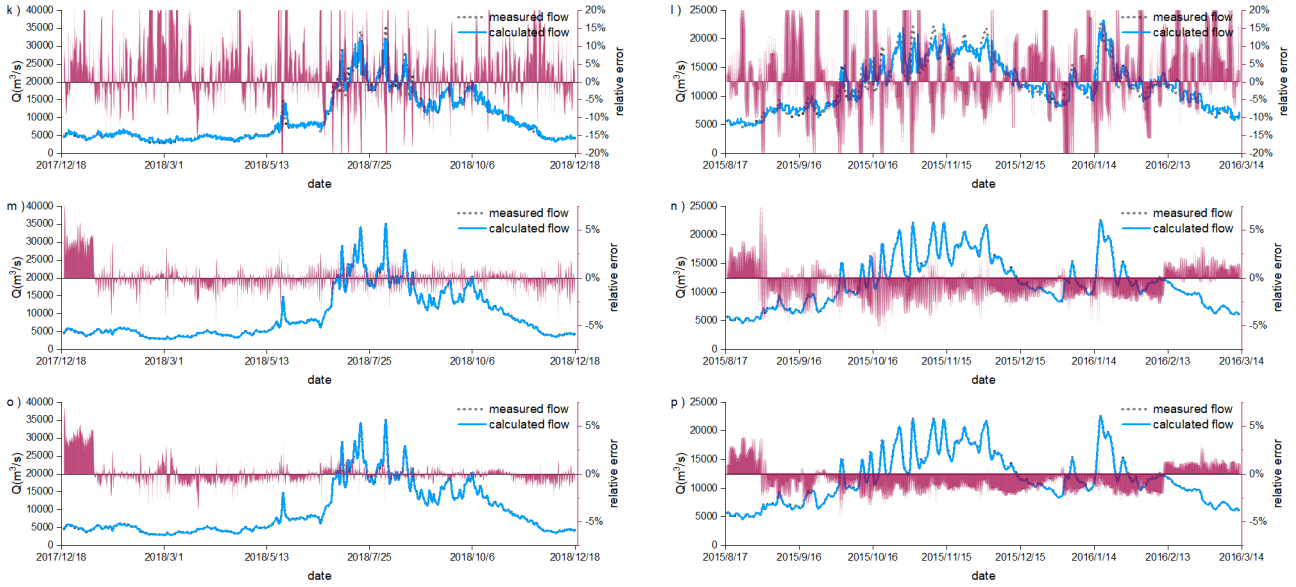

Figure 6. Computation results of eight deep learning networks in Zhutuo (the red bar in these figures represent relative error at different times and the value is determined by the right vertical axis). (**a**,**b**) BiGRU's performances on the training dataset and testing dataset respectively; (**c**,**d**) BiLSTM's performances on the training dataset and testing dataset, respectively; (**e**,**f**) BiRNN's performances on the training dataset and testing dataset, respectively; (**g**,**h**) GRU's performances on the training dataset and testing dataset, respectively; (**i**,**j**) LSTM's performances on the training dataset and testing dataset, respectively; (**k**,**l**) RNN's performances on the training dataset and testing dataset, respectively; (**m**,**n**) Seq2Seq's performances on the training dataset and testing dataset, respectively; (**o**,**p**) Sq2Seq attention's performances on the training dataset and testing dataset, respectively.

The results indicated that most of the deep learning models performed exceptionally well. In particular, BiGRU and BiLSTM exhibited outstanding learning ability (as shown in Figure 7), with relative errors under 2.5% in the entire year. The learning ability of GRU and LSTM was also impressive, with relative errors under 5%. The Seq2Seq and Seq2Seq attention networks performed reasonably well, with relative errors mostly under 5%. However, the RNN and BiRNN networks' learning abilities were relatively weak compared to other networks, with most of their relative errors exceeding 10%.

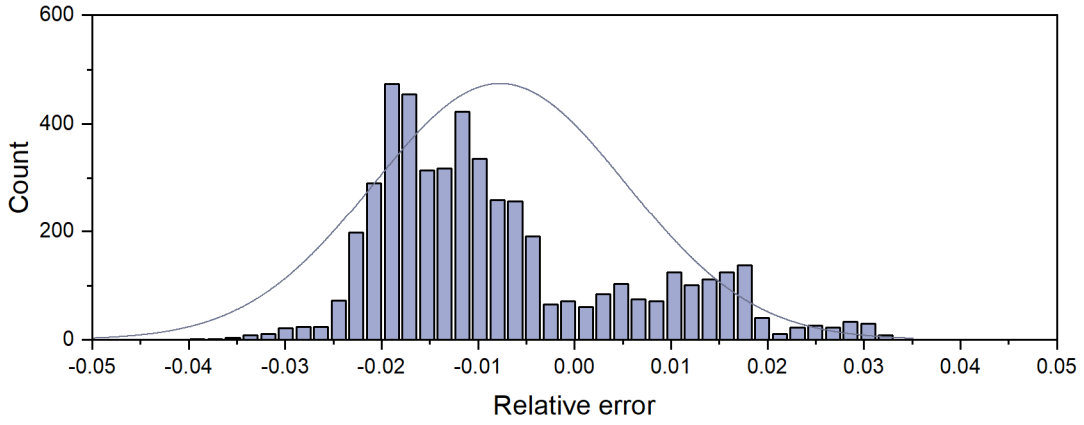

Figure 7. The histogram plot of BiGRU.

The test set consists of data that has not been trained, and the performance of the models on test set determined their practical application ability. BiGRU, BiLSTM, GRU, LSTM, and Seq2Seq attention performed exceptionally well, with relative errors under 5%, while RNN exhibited relatively poor performance, with many relative errors exceeding

20%. BiRNN was an improvement over RNN, but its relative errors were still over 10%. Notably, Seq2Seq attention, GRU, and LSTM exhibited similar performances to BiGRU and BiLSTM, even though they did not completely learn the training dataset.

Furthermore, the test dataset spanned seven months, indicating that deep learning models such as BiGRU could estimate hourly river flow based on water level continuously for a long-term period. However, Figure 6 also revealed that the deep learning models tended to underestimate the measured flow during the dry season and overestimate it during the wet season. This suggested that the models might have difficulty accurately identifying periods of rising and falling water levels, or that some of the discrepancies might be due to measurement errors.

3.2. Estimated Results in Different Parameters

This study employs BiGRU networks to investigate the effects of hyperparameters on network performance, due to its satisfactory performance and moderate parameter count. Table 1 presents the combinations of four hyperparameters, namely batch size, time step, number of neurons, and number of layers. The nine groups of hyperparameters were created by varying these parameters across different ranges. The BiGRU network was trained for 24 h with each of the nine groups of hyperparameters, and then evaluated on the test dataset using four evaluation metrics: relative mean absolute percentage error (RMSE), Nash–Sutcliffe efficiency coefficient (NSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

Table 1. Nine types of combinations of the four hyperparameters *.

Hyperparameters	a	b	c	d	e	f	g	h	i
Batch size	1	1	1	1	1	1	1	12	24
Time step	72	72	48	72	72	96	72	72	72
Number of neurons	128	64	128	128	128	128	256	128	128
Number of layers	2	2	2	4	6	2	2	2	2

Note: * One bi-directional layer contains both forward and backward hidden states. So, here we consider a bidirectional layer as two layers.

The results presented in Figure 8 showed that the relative errors of groups a–i on the test dataset were within ±5%, indicating that BiGRU performed well despite the variations in hyperparameters. Table 2 suggests that increasing batch size can enhance network performance, as groups h and i exhibited improvements of 1.86% and 1.39% in RMSE and MAE compared to group a. Furthermore, group i showed a 3.74% and 3.92% improvement in RMSE and MAE, respectively, compared to group a. In contrast, variations in time step had minimal effects on network performance. Group c, a, and f, with 48, 72, and 96 time steps, respectively, exhibited only slight differences.

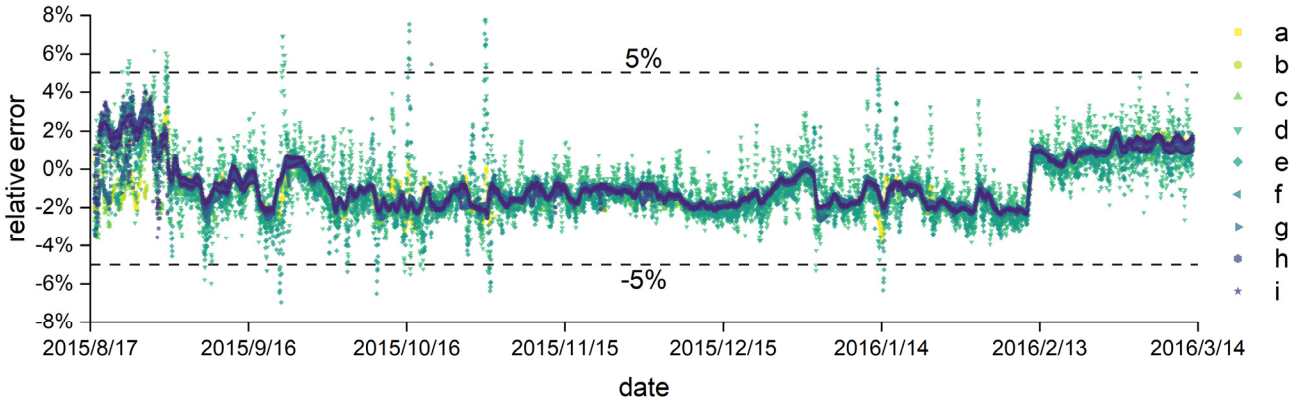

Figure 8. Relative errors of nine combinations of hyperparameters in the test dataset of Zhutuo.

Table 2. Comprehensive performance results of nine types of hyperparameters.

Evaluation Metrics	a	b	c	d	e	f	g	h	i
RMSE	266.008	258.448	268.750	267.236	269.780	266.016	256.950	261.046	256.058
NSE	0.9971	0.9973	0.9971	0.9971	0.9970	0.9971	0.9973	0.9972	0.9973
MAE	216.089	210.169	219.553	210.205	213.596	215.500	207.430	213.075	207.624
MAPE	0.0369	0.0355	0.0355	0.0352	0.0336	0.0369	0.0360	0.0357	0.0355

Regarding the number of neurons, the comparison between groups b, a, and g indicated that all four metrics in groups b and g were better than those in group a, even with fewer neurons (64 and 256 compared to 128). Group b exhibited a 2.84% and 2.74% improvement in RMSE and MAE, respectively, compared to group a, while group g showed a 3.41% and 4.01% improvement in RMSE and MAE, respectively, compared to group a. Finally, the number of layers did not have a significant impact on network performance.

Table 3 ranks the performance of groups a–i using the evaluation metrics, where the best group in each metric receives a score of 1, and the worst group receives a score of 9. The total scores were calculated, and the lower the score, the better the performance. Group i demonstrated the best overall performance, with a total score of 7, using a batch size of 24, time step of 72, number of neurons of 128, and 2 layers. Notably, the differences in the four evaluation metrics between the best-performing group and the worst-performing group were less than 5%, indicating that the worst-performing group of hyperparameters still had the ability to generate good performances.

Table 3. Scores of nine types of hyper-parameter *.

Evaluation Metrics	a	b	c	d	e	f	g	h	i
RMSE	5	3	8	7	9	6	2	4	1
NSE	5	1	5	5	9	5	1	4	1
MAE	8	3	9	4	6	7	1	5	2
MAPE	8	3	3	2	1	8	7	6	3
Total scores	26	10	25	18	25	26	11	19	7

Note: * This table ranks the performance of groups a–i using the evaluation metrics, where the best group in each metric receives a score of 1 and the worst group receives a score of 9.

3.3. Estimated Results in Different Datasets

In this study, we investigate the performance of the BiGRU network in computing discharge at six different stations on the Yangtze River basin. The stations include Gaochang, Fushun, Panzhihua, Sanduizi, Wudongde, and Zhutuo. The network parameters used are the same as described in Section 3.2. The results are presented in Tables 4 and 5, showing that the BiGRU network performed well at all six stations. However, there are some variations in the performance between the stations. The network achieved the highest total score of 4 at Sanduizi and the lowest score of 21 at Panzhihua, as calculated using the method described in Section 3.2.

Table 4. Comprehensive performance results of six stations.

Evaluation Metrics	Gaochang	Fushun	Panzhihua	Sanduizi	Wudongde	Zhutuo
RMSE	154.084	77.161	214.67	16.257	98.836	266.009
NSE	0.9898	0.9800	0.9678	0.9999	0.9984	0.9971
MAE	70.546	47.760	94.089	12.443	80.534	216.089
MAPE	0.0536	0.1272	0.0548	0.004	0.0276	0.0370

Table 5. Scores of the performances on six stations *.

Evaluation Metrics	Gaochang	Fushun	Panzhihua	Sanduizi	Wudongde	Zhutuo
RMSE	4	2	5	1	3	6
NSE	4	5	6	1	2	3
MAE	3	2	5	1	4	6
MAPE	4	6	5	1	2	3
Total scores	15	15	21	4	11	18

Note: * The calculation method of the scores in this table is the same as in Table 3.

To further evaluate the network's performance, we plotted the correlations between estimated discharges and observed discharges for all six stations in Figure 9. The results show that the BiGRU network performed best at Sanzuizi, Wudongde, and Zhutuo, with NSE values of 0.9999, 0.9984, and 0.9971, respectively. Although the network also performed well in Gaochang, Fushun, and Panzhihua, the computation accuracy at these stations was lower than at the other sites. The differences in the network's performance between stations may be due to variations in hydraulic properties and the impact of hydraulic engineering structures. For instance, the construction or operation of dams can significantly affect local water levels and disrupt the relationship between discharge and water level. It can be observed in Panzhihua station that the water level nearly stays the same when downstream dam regulation of water resources occurs. This stability of water level serves as a key factor contributing to the close proximity of estimated river discharge values within specific timeframes (refer to Figure 9c).

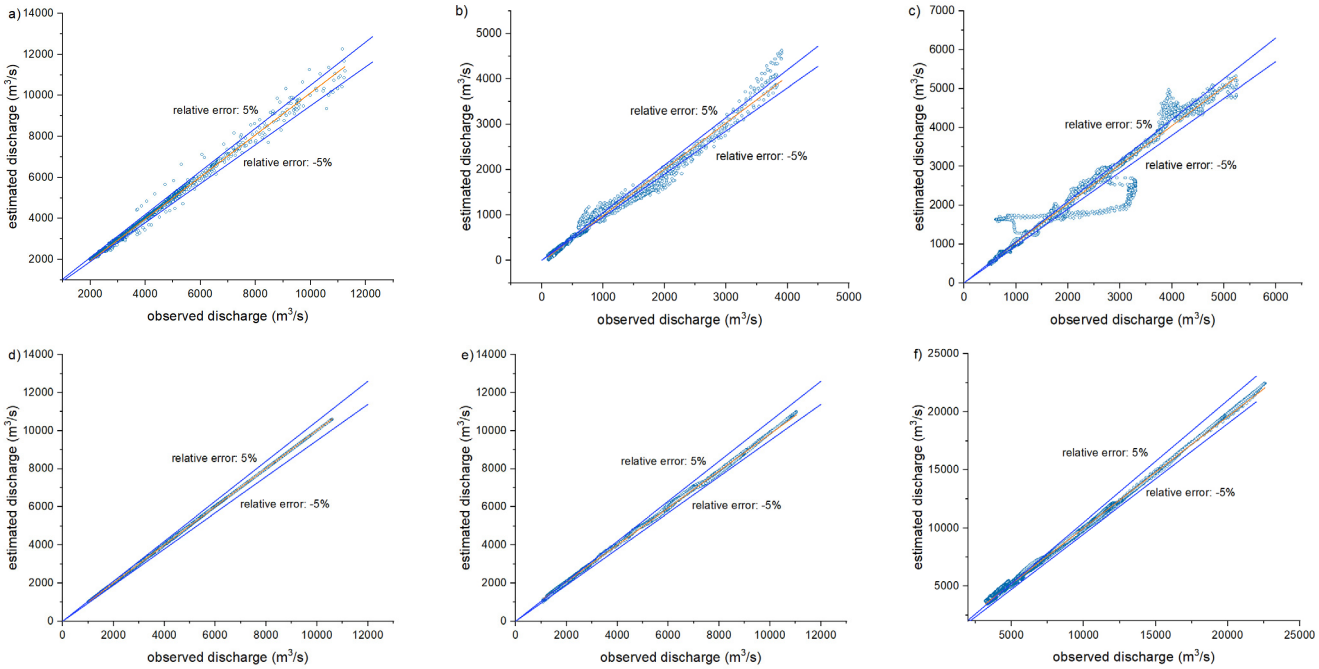

Figure 9. Relation between estimated discharge and observed discharge in six stations. (**a**) Gaochang, (**b**) Fushun, (**c**) Panzhihua, (**d**) Sanduizi, (**e**) Wudongde, and (**f**) Zhutuo.

Overall, these results demonstrate the potential of deep learning in discharge computation, but also highlight the importance of considering variations in hydraulic properties and engineering structures when applying deep learning methods to different stations.

4. Discussion

Based on the experimental results presented earlier, it is evident that the data-driven based method can effectively compute river flow directly from water level data. This

approach demonstrates remarkable accuracy and stability across long-term hourly time series. However, it remains paramount to ascertain the depth of understanding that deep learning networks have acquired concerning the underlying river flow computation process derived from water level data. To this end, this chapter conducts a white noise examination on the absolute errors. The identification of white noise within these errors would signify that deep learning networks have successfully assimilated the maximum pertinent information from the river flow and water level data. The results of this white noise analysis are visually represented in Figure 10. This figure shows that the absolute errors in all six stations exhibit a characteristic pattern of white noise, which cannot be predicted or modeled. This pattern is commonly referred to as a random walk, which indicates that the errors are purely random and based on the data preceding them. This conclusion is supported by the fact that the ACF (auto correlation function) of all the absolute error series decrease slowly, while the PACF (partial autocorrelation function) drops sharply from the third point, which is a well-known phenomenon of random walk.

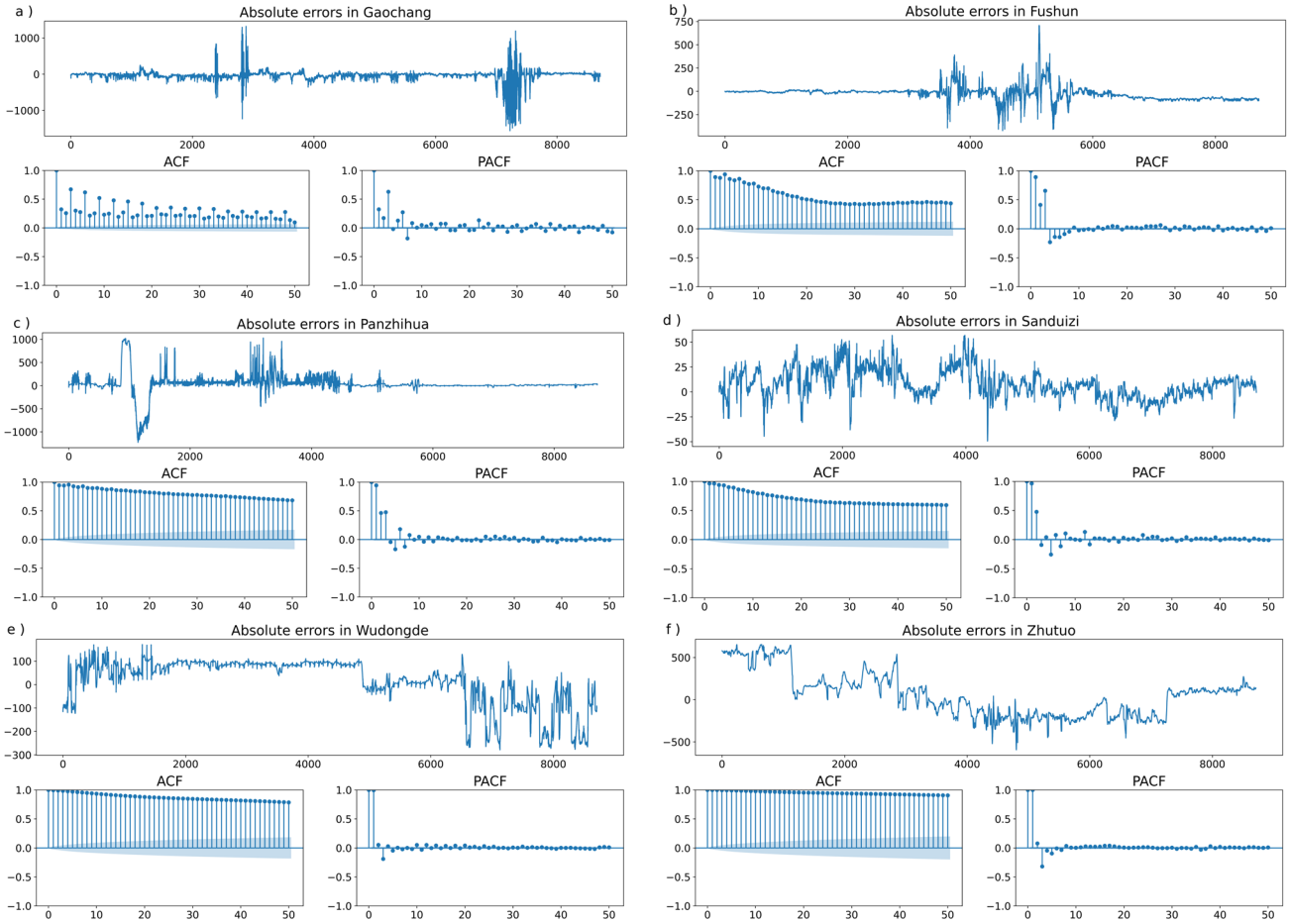

Figure 10. White noise examination of six stations. (**a**) Gaochang, (**b**) Fushun, (**c**) Panzhihua, (**d**) Sanduizi, (**e**) Wudongde, and (**f**) Zhutuo.

In light of these phenomena, it can be concluded that the deep learning network has successfully gleaned valuable insights from both river flow and water level data. This proficiency equips it with the capacity to accurately compute river flow solely based on water level data.

The outcomes of this examination are robust, affirming the data-driven based method's competence in the computation of river flow from water level data. Notably, the BiGRU network, classified as a variant of RNN, generates each data point contingent upon its predecessor. This dynamic may contribute to the discerned random walk pattern observed

in the absolute errors. Furthermore, it is a well-established fact that measured data often harbors systematic errors, akin to a form of white noise. Nonetheless, despite these complexities, deep learning networks consistently deliver commendable performance, providing precise estimates. This underscores their efficacy in addressing the challenge of direct river flow computation from water level data. Unlike previous usage of deep learning to compute short-term river discharge that need other input parameters besides water level, such as rainfall [26], the approach used in this paper proved that deep learning has the potential to accurately calculate long-term river discharge purely based on water level. However, it remains imperative to acknowledge the presence of errors and advocate for ongoing refinement of the models and methodologies employed to further enhance accuracy and mitigate potential biases.

5. Conclusions

In summary, we applied deep learning networks to compute long-term river flow from water level measurements. Our findings strongly indicate that deep learning networks are capable of accurately computing river discharge with only water surface elevation data. This approach not only enhances convenience but also offers the potential for cost reduction in river flow measurement. We compared the performance of eight different deep learning networks, and it shows that most of them perform well. Our exploration of the effects of different hyperparameters on the BiGRU model showed that the relative errors of different hyperparameters on the test dataset were within $\pm 5\%$, indicating that BiGRU performed well despite the variations in hyperparameters. Applying the BiGRU model to six gauging stations along the Yangtze River showed that it can be effectively applied to all six stations with high accuracy in computing river flow, but the network's performances may be affected by the impact of hydraulic engineering structures and various hydraulic properties, which is valuable for furthering research. Overall, with this data-driven based approach, river discharge can be accurately, objectively, and quickly computed directly from water surface elevation, which is of practical value for flood protection and water resources management.

Author Contributions: W.L. (First Author): Methodology, Investigation, Formal Analysis, Supervision, Writing—Original Draft, Data Curation, Funding Acquisition; P.Z. (Corresponding Author): Conceptualization, Methodology, Software, Formal Analysis, Writing—Original Draft Conceptualization, Visualization; D.J.: Conceptualization, Investigation, Methodology, Formal Analysis, Resources, Writing—Review and Editing; X.Q.: Writing—Review and Editing, Software, Visualization; H.D.: Resources, Supervision, Conceptualization, Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China (No. 2022YFC3203900) and the National Natural Science Foundation of China (No. U2040220).

Data Availability Statement: Not applicable.

Acknowledgments: Special thanks are given to the anonymous reviewers and editors for their constructive comments.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Tarpanelli, A.; Paris, A.; Sichangi, A.W.; O'Loughlin, F.; Papa, F. Water Resources in Africa: The Role of Earth Observation Data and Hydrodynamic Modeling to Derive River Discharge. *Surv. Geophys.* **2023**, *44*, 97–122. [CrossRef]
2. Xia, M.; Craig, P.M.; Schaeffer, B.; Stoddard, A.; Liu, Z.; Peng, M.; Zhang, H.; Wallen, C.M.; Bailey, N.; Mandrup-Poulsen, J. Influence of Physical Forcing on Bottom-Water Dissolved Oxygen within Caloosahatchee River Estuary, Florida. *J. Environ. Eng.* **2010**, *136*, 1032–1044. [CrossRef]
3. Xia, M.; Craig, P.M.; Wallen, C.M.; Stoddard, A.; Mandrup-Poulsen, J.; Peng, M.; Schaeffer, B.; Liu, Z. Numerical Simulation of Salinity and Dissolved Oxygen at Perdido Bay and Adjacent Coastal Ocean. *J. Coast. Res.* **2011**, *27*, 73. [CrossRef]

4. Chen, J.; Weisberg, R.H.; Liu, Y.; Zheng, L. Hillsborough Bay Inflow Modification Study: An Application of the Tampa Bay Coastal Ocean Model. *Estuar. Coast. Shelf Sci.* **2023**, *281*, 108213. [CrossRef]
5. Cheng, Z.; Lee, K.; Kim, D.; Muste, M.; Vidmar, P.; Hulme, J. Experimental Evidence on the Performance of Rating Curves for Continuous Discharge Estimation in Complex Flow Situations. *J. Hydrol.* **2019**, *568*, 959–971. [CrossRef]
6. Chen, Y.-C.; Yang, T.-M.; Hsu, N.-S.; Kuo, T.-M. Real-Time Discharge Measurement in Tidal Streams by an Index Velocity. *Environ. Monit. Assess.* **2012**, *184*, 6423–6436. [CrossRef] [PubMed]
7. Le Coz, J.; Pierrefeu, G.; Paquier, A. Evaluation of River Discharges Monitored by a Fixed Side-Looking Doppler Profiler. *Water Resour. Res.* **2008**, *44*. [CrossRef]
8. Lee, K.; Firoozfar, A.R.; Muste, M. Technical Note: Monitoring of Unsteady Open Channel Flows Using the Continuous Slope-Area Method. *Hydrol. Earth Syst. Sci.* **2017**, *21*, 1863–1874. [CrossRef]
9. Lee, K.; Muste, M. Refinement of the Fread Method for Improved Tracking of Stream Discharges during Unsteady Flows. *J. Hydraul. Eng.* **2017**, *143*, 06017003. [CrossRef]
10. Muste, M.; Lee, K.; Kim, D.; Bacotiu, C.; Oliveros, M.R.; Cheng, Z.; Quintero, F. Revisiting Hysteresis of Flow Variables in Monitoring Unsteady Streamflows. *J. Hydraul. Res.* **2020**, *58*, 867–887. [CrossRef]
11. Gonçalves, G.M.S.; Bartels, G.K.; Lima, L.S.; Boeira, L.d.S.; Collares, G.L. Continuous Discharge Monitoring of the Mirim-São Gonçalo System by the Index Velocity Rating Curve Method. *J. Hydroinformatics* **2023**, *25*, 20–35. [CrossRef]
12. Pappenberger, F.; Matgen, P.; Beven, K.J.; Henry, J.-B.; Pfister, L.; Fraipont, P. Influence of Uncertain Boundary Conditions and Model Structure on Flood Inundation Predictions. *Adv. Water Resour.* **2006**, *29*, 1430–1449. [CrossRef]
13. Gensen, M.R.A.; Warmink, J.J.; Huthoff, F.; Hulscher, S.J.M.H. Feedback Mechanism in Bifurcating River Systems: The Effect on Water-Level Sensitivity. *Water* **2020**, *12*, 1915. [CrossRef]
14. Berends, K.D.; Straatsma, M.W.; Warmink, J.J.; Hulscher, S.J.M.H. Uncertainty Quantification of Flood Mitigation Predictions and Implications for Interventions. *Nat. Hazards Earth Syst. Sci.* **2019**, *19*, 1737–1753. [CrossRef]
15. Le Coz, J.; Renard, B.; Bonnifait, L.; Branger, F.; Le Boursicaud, R. Combining Hydraulic Knowledge and Uncertain Gaugings in the Estimation of Hydrometric Rating Curves: A Bayesian Approach. *J. Hydrol.* **2014**, *509*, 573–587. [CrossRef]
16. Di Baldassarre, G.; Claps, P. A Hydraulic Study on the Applicability of Flood Rating Curves. *Hydrol. Res.* **2011**, *42*, 10–19. [CrossRef]
17. Lang, M.; Pobanz, K.; Renard, B.; Renouf, E.; Sauquet, E. Extrapolation of Rating Curves by Hydraulic Modelling, with Application to Flood Frequency Analysis. *Hydrol. Sci. J.* **2010**, *55*, 883–898. [CrossRef]
18. Yilmaz, A.G.; Muttil, N. Runoff Estimation by Machine Learning Methods and Application to the Euphrates Basin in Turkey. *J. Hydrol. Eng.* **2014**, *19*, 1015–1025. [CrossRef]
19. Hu, C.; Wu, Q.; Li, H.; Jian, S.; Li, N.; Lou, Z. Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *Water* **2018**, *10*, 1543. [CrossRef]
20. Wee, W.J.; Zaini, N.B.; Ahmed, A.N.; El-Shafie, A. A Review of Models for Water Level Forecasting Based on Machine Learning. *Earth Sci. Inform.* **2021**, *14*, 1707–1728. [CrossRef]
21. Yang, M.; Yang, Q.; Shao, J.; Wang, G.; Zhang, W. A New Few-Shot Learning Model for Runoff Prediction: Demonstration in Two Data Scarce Regions. *Environ. Model. Softw.* **2023**, *162*, 105659. [CrossRef]
22. Pokharel, S.; Roy, T.; Admiraal, D. Effects of Mass Balance, Energy Balance, and Storage-Discharge Constraints on LSTM for Streamflow Prediction. *Environ. Model. Softw.* **2023**, *166*, 105730. [CrossRef]
23. Ni, J.; Liu, R.; Li, Y.; Tang, G.; Shi, P. An Improved Transfer Learning Model for Cyanobacterial Bloom Concentration Prediction. *Water* **2022**, *14*, 1300. [CrossRef]
24. Shan, S.; Ni, H.; Chen, G.; Lin, X.; Li, J. A Machine Learning Framework for Enhancing Short-Term Water Demand Forecasting Using Attention-BiLSTM Networks Integrated with XGBoost Residual Correction. *Water* **2023**, *15*, 3605. [CrossRef]
25. Wongburi, P.; Park, J.K. Prediction of Wastewater Treatment Plant Effluent Water Quality Using Recurrent Neural Network (RNN) Models. *Water* **2023**, *15*, 3325. [CrossRef]
26. Liu, G.; Tang, Z.; Qin, H.; Liu, S.; Shen, Q.; Qu, Y.; Zhou, J. Short-Term Runoff Prediction Using Deep Learning Multi-Dimensional Ensemble Method. *J. Hydrol.* **2022**, *609*, 127762. [CrossRef]
27. Matsui, K.; Shirai, H.; Kageyama, Y.; Yokoyama, H.; Asano, M. Estimating Water Quality through Neural Networks Using Terra ASTER Data, Water Depth, and Temperature of Lake Hachiroko, Japan. *Environ. Model. Softw.* **2023**, *159*, 105584. [CrossRef]
28. Han, H.; Morrison, R.R. Improved Runoff Forecasting Performance through Error Predictions Using a Deep-Learning Approach. *J. Hydrol.* **2022**, *608*, 127653. [CrossRef]
29. Wang, Y.; Huang, Y.; Xiao, M.; Zhou, S.; Xiong, B.; Jin, Z. Medium-Long-Term Prediction of Water Level Based on an Improved Spatio-Temporal Attention Mechanism for Long Short-Term Memory Networks. *J. Hydrol.* **2023**, *618*, 129163. [CrossRef]
30. Xiang, Z.; Yan, J.; Demir, I. A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning. *Water Resour. Res.* **2020**, *56*. [CrossRef]
31. Lu, J.; Gu, J.; Han, J.; Xu, J.; Liu, Y.; Jiang, G.; Zhang, Y. Evaluation of Spatiotemporal Patterns and Water Quality Conditions Using Multivariate Statistical Analysis in the Yangtze River, China. *Water* **2023**, *15*, 3242. [CrossRef]
32. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.

33. Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv* **2014**, arXiv:1409.1259.
34. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
35. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2016**, arXiv:1409.0473.
36. Samantaray, S.; Sahoo, P.; Sahoo, A.; Satapathy, D.P. Flood Discharge Prediction Using Improved ANFIS Model Combined with Hybrid Particle Swarm Optimisation and Slime Mould Algorithm. *Environ. Sci. Pollut. Res. Int.* **2023**, *30*, 83845–83872. [CrossRef]
37. Samantaray, S.; Sahoo, A. Prediction of Flow Discharge in Mahanadi River Basin, India, Based on Novel Hybrid SVM Approaches. *Environ. Dev. Sustain.* **2023**, 1–25. [CrossRef]
38. Samantaray, S.; Ghose, D.K. Sediment Assessment for a Watershed in Arid Region via Neural Networks. *Sādhanā* **2019**, *44*, 219. [CrossRef]
39. Samantaray, S.; Ghose, D.K. Prediction of S12-MKII Rainfall Simulator Experimental Runoff Data Sets Using Hybrid PSR-SVM-FFA Approaches. *J. Water Clim. Chang.* **2022**, *13*, 707–734. [CrossRef]