



---

*Research article*

## Uncertainty prediction of wind speed based on improved multi-strategy hybrid models

Xinyi Xu<sup>1</sup>, Shaojuan Ma<sup>1,2,\*</sup> and Cheng Huang<sup>1</sup>

<sup>1</sup> School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China

<sup>2</sup> Ningxia Key Laboratory of Intelligent Information and Big Data Processing, North Minzu University, Yinchuan 750021, China

\* **Correspondence:** Email: [sjma@nmu.edu.cn](mailto:sjma@nmu.edu.cn).

**Abstract:** Accurate interval prediction of wind speed plays a vital role in ensuring the efficiency and stability of wind power generation. Due to insufficient traditional wind speed interval prediction methods for mining nonlinear features, in this paper, a novel interval prediction method was proposed by combining improved wavelet threshold and deep learning (BiTCN-BiGRU) with the nutcracker optimization algorithm (NOA). First, NOA was used to optimize the wavelet transform (WT) and BiTCN-BiGRU. Second, we applied NOA-WT to smooth the wind speed data. Then, to capture nonlinear features of time series, phase space reconstruction (PSR) was utilized to identify chaotic characteristics of the processed data. Finally, the NOA-BiTCN-BiGRU model was built to perform wind speed interval prediction. Under the same hyperparameters and network structure settings, a comparison with other deep learning methods showed that the prediction interval coverage probability (PICP) and prediction interval mean width (PIMW) of NOA-WT-BiTCN-BiGRU model achieves the best balance, with good prediction accuracy and generalization performance. This research can provide reference and guidance for nonlinear time-series interval prediction in the real world.

**Keywords:** wind speed interval prediction; chaotic time series; nutcracker optimization algorithm; wavelet threshold; BiTCN-BiGRU

---

### 1. Introduction

Wind energy, as a renewable and clean energy source, is increasingly favored by countries worldwide and has become a key focus in the development of new energy technologies [1, 2]. However, the inherent randomness, volatility, and intermittency of wind speed time series data could lead to unstable and uncontrollable power generation, which can threaten the stability of power grid operations [3]. To improve the accuracy of wind speed interval prediction and reduce the impact of noise, Donoho

proposed the wavelet threshold denoising method based on wavelet transform in 1995 [4, 5]. Wu et al. [6] applied the wavelet threshold to filter wind speed data and combined it with a multivariate long short-term memory (LSTM) model to forecast with better accuracy. Wavelet thresholding was applied to denoise wind power data by Lian et al. [7], together with improved support vector machine (SVM) to determine the parameters for the interval prediction. Karijadi et al. [8] decomposed the wind speed data into several sub-sequences by wavelet thresholds, and predicted using LSTM with higher prediction accuracy. However, the discontinuity of the hard threshold function [9] and the derivative of the soft threshold function [10] limit further applications for the wavelet threshold method [11].

In order to overcome the shortcomings of traditional soft and hard thresholding in data denoising, Peng et al. [12] developed a wind speed prediction model that integrates wavelet soft thresholding filtering and gate recurrent unit (GRU), which improves model accuracy by eliminating redundant information and optimizing GRU parameters. A new fixed threshold formula was designed by [13] introducing the logarithmic function of wavelet decomposition layers, which can significantly improve the signal-to-noise ratio. Wang et al. [14] developed an improved wavelet threshold function that combines the advantages of hard and soft threshold functions to reduce errors. Qian [15] proposed a continuous and differentiable wavelet threshold combined with median filtering, which demonstrated a good denoising effect in experiments. Liao and collaborators [16] incorporated an improved threshold function with a variable parameter factor, controlled by the critical scale, to achieve more effective denoising effects. Qiao et al. [17] introduced two adjustment factors into the wavelet threshold function, which can ensure the continuity of the function at the threshold point and solve the deviation problem of wavelet coefficients. To solve the problem of improper adjustment factor settings leading to excessive smoothing of WT data in the above results, researchers in [18] used a genetic algorithm to perform parameter tuning for the wavelet threshold. Besides this solution, other advanced optimization algorithms are worth exploring to adjust wavelet threshold parameters, such as the sparrow search algorithm (SSA), the nutcracker optimizer algorithm (NOA), and others. Due to the nonstationarity and volatility of wind speed time series, point prediction is insufficient to accurately reflect the uncertainty of wind speed. Interval prediction provides an effective solution for estimating the reliability and error range of wind speed predictions [19]. At present, there are two types of interval prediction. The first type is based on point prediction results, further constructing prediction intervals. Li et al. [20] optimized the parameters of the least squares support vector machine (LSSVM) for the prediction interval based on model point prediction. Zhang et al. [21] constructed the improved particle swarm optimization (PSO) to optimize the radial basis function (RBF) network and estimate wind speed range based on point prediction results. Gan et al. [22] proposed a novel wind speed interval prediction model based on temporal convolutional networks (TCN), which can directly generate prediction intervals. However, this interval prediction method overly relies on the performance of point prediction models, which is easily affected by noise or outliers.

The second type of interval prediction is based on probability and statistical fitting of prediction intervals. The parameterization method constructs a prediction interval by fitting a certain probability distribution function to generate the distribution probability of trajectory points. Liu et al. [23] employed variational Bayesian inference to obtain an approximate posterior parameter distribution of the model and used a spatiotemporal neural network to obtain uncertainty estimates. A model combining the beta distribution function and LSTM neural network was proposed by Yuan et al. [24] to improve interval prediction performance. Pei et al. [25] obtained interval prediction values by com-

paring and selecting prediction results with probability distribution density functions. However, the parameter method relies on pre-set data distributions and lacks flexibility when dealing with complex and irregular data, which is difficult to adapt to the multimodality or asymmetry of the data.

Nonparametric methods can be used for interval prediction without making any assumptions about the distribution. Zhang et al. [26] proposed a wind power interval prediction method based on non-parametric kernel density estimation, which can obtain the shortest prediction interval at different confidence levels. Wang et al. [27] implemented a new ensemble probability prediction strategy combined with quantile regression (QR) and bidirectional long short-term memory (BiLSTM) to explore ensemble probability prediction. Peng et al. [28] developed a new neural network prediction model that combines LSTM with QR, which has good accuracy and reliability in predicting intervals and probabilities. Although QR is a flexible and efficient prediction method, prediction results depend on the selected regression model and require reasonable model settings and parameter adjustments. Wang et al. [29] proposed a new prediction model combining improved algorithms and QR for probability interval prediction. To optimize parameters by algorithms, the coverage and accuracy of interval prediction have been significantly improved, with higher robustness and narrower average bandwidth.

Through the above analysis, it was found that each model has its own characteristics in wind speed interval prediction. On one hand, the improved wavelet threshold function addresses the issues of discontinuity and nondifferentiability in traditional threshold functions. On the other hand, deep learning models can effectively capture the nonlinear characteristics of time series data. Although the WT and the deep learning models have some advantages, the parameter settings for both models is a prominent issue that can lead to excessive data smoothing and a decrease in model generalization performance, resulting in worse model prediction accuracy, as shown in Table 1. To solve these problems, in this paper, the parameters are optimized by the intelligent algorithm based on wind data, and then the prediction is carried out.

**Table 1.** Literature review.

Model&Method	Main content	Defect
Wavelet threshold	Integrated threshold function [12]	Loss of information and nonlinear features.
	Introducing logarithmic function [13]	
	Combined threshold function [14]	
	A continuous and differentiable wavelet threshold [15]	
	Introducing a variable factor [16]	
Interval prediction	Introducing two adjustment factors [17]	A decrease in the generalization performance of interval prediction.
	Threshold function for algorithm optimization [18]	
	Kernel density estimation [26]	
	QR-BiLSTM [27]	
Interval prediction	QR-LSTM [28]	A decrease in the generalization performance of interval prediction.
	QR-MMOTA-BMs [29]	

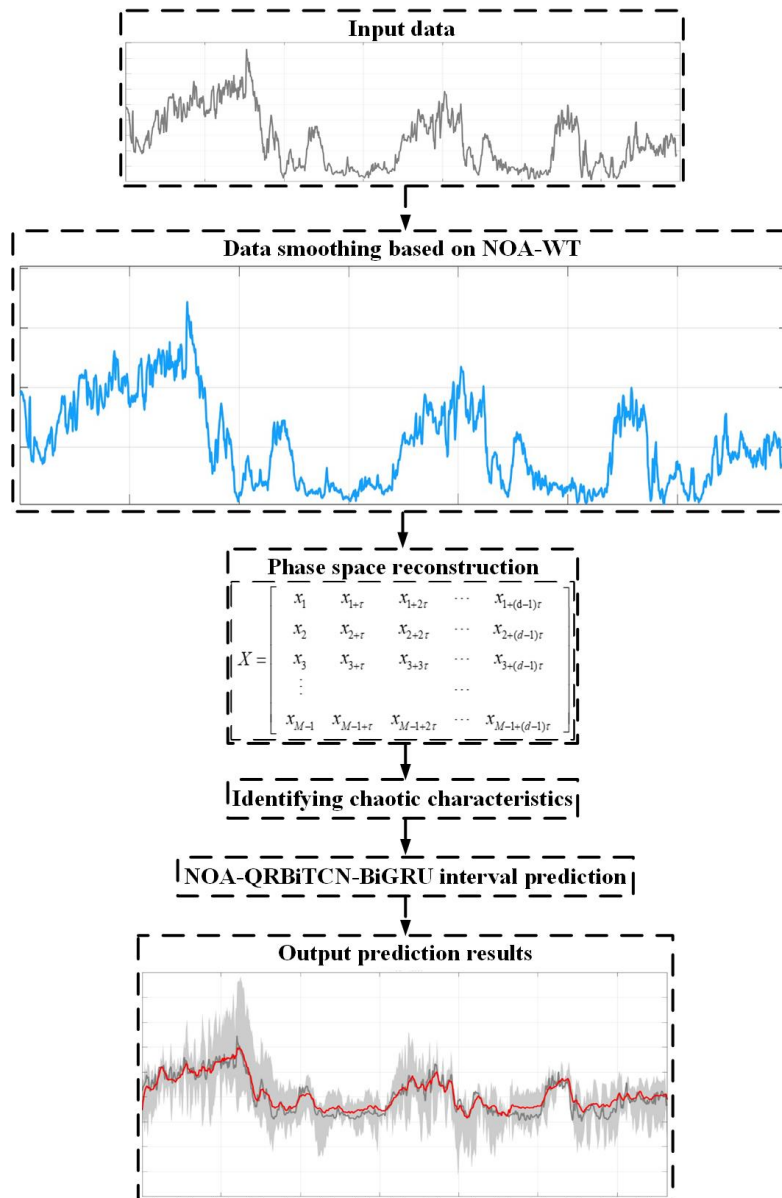
The innovation and main contributions of this article are as follows:

- 1) We optimize the WT and deep learning models for better parameter settings using NOA, which has demonstrated excellent performance among highly-cited algorithms.
- 2) In order to capture the nonlinearity of the data to choose the more suitable prediction method, we

apply PSR to identify the chaotic characteristics of the data.

3) We verify the progressiveness of the proposed model system by setting different combination models, ablation experiments, algorithm comparisons and other experiments.

The remaining structure of the article is as follows: Section 2 provides a detailed introduction to intelligent algorithm optimization of WT and deep learning models. In Section 3, we conduct interval prediction based on two datasets. In Section 4, experimental verification and comparison are performed, and the results are analyzed. Section 5 draws conclusions. The structure of the article is shown in Figure 1.



**Figure 1.** Article framework.

## 2. Model construction

### 2.1. NOA-WT model

#### 2.1.1. Nutcracker optimization algorithm

The nutcracker optimization algorithm (NOA) was proposed by Mohamed Abdel Baset et al. [30]. The authors evaluated the NOA using 23 classic benchmark functions, CEC2014 test set, CEC2017 test set, CEC2020 test set, and five engineering problems. Compared with the three existing types of optimization algorithms, the experimental results show that NOA ranks first and has the best overall performance.

#### 2.1.2. Wavelet transform

Wavelet threshold functions are divided into hard threshold functions and soft threshold functions. It is generally believed that wavelet coefficients below the threshold are generated by noise, while wavelet coefficients above the threshold are generated by effective data. The calculation formula is as follows:

Hard threshold function:

$$\eta(w_{j,k}, \lambda) = \begin{cases} w_{j,k}, & \text{if } |w_{j,k}| \geq \lambda \\ 0, & \text{if } |w_{j,k}| < \lambda \end{cases}.$$

Soft threshold function:

$$\eta(w_{j,k}, \lambda) = \begin{cases} \text{sgn}(w_{j,k}) \cdot (|w_{j,k}| - \lambda), & \text{if } |w_{j,k}| \geq \lambda \\ 0, & \text{if } |w_{j,k}| < \lambda \end{cases}.$$

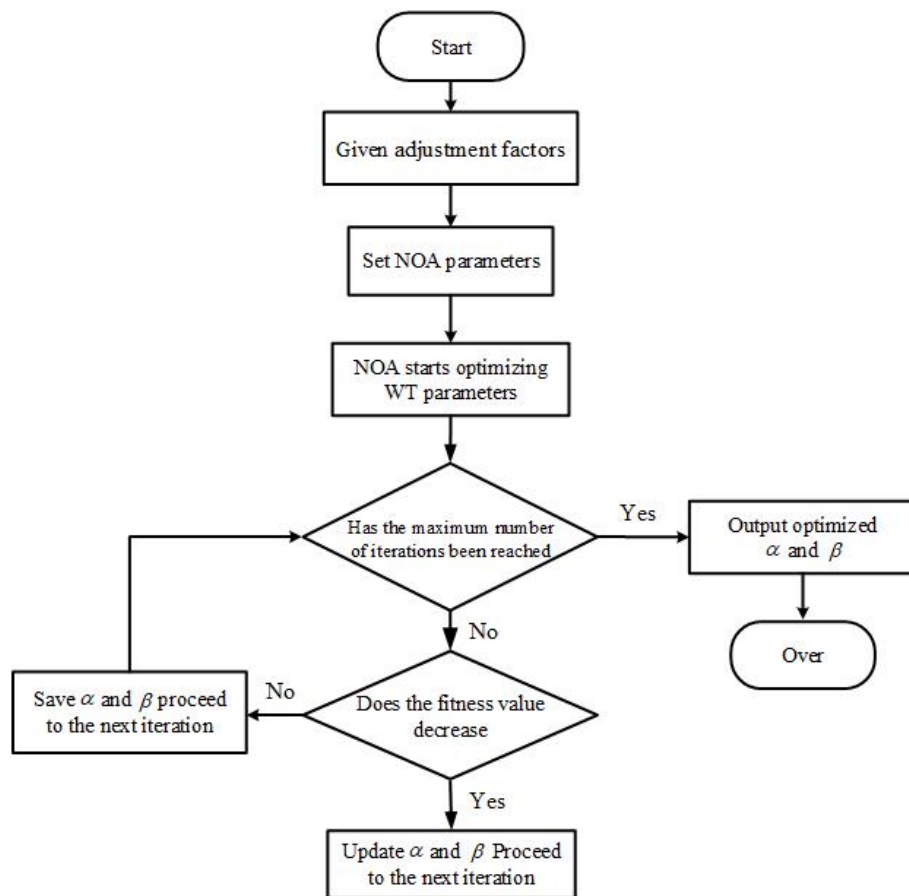
The improved wavelet threshold function [17] used in this paper is as follows:

$$W_{j,k} = \begin{cases} \text{sgn}(w_{j,k}) \cdot \left| w_{j,k} - \frac{\lambda}{\beta \sqrt{|w_{j,k}| - |\lambda|^\beta + 1}} \cdot \frac{1}{e^{\alpha \sqrt{|w_{j,k}| - |\lambda|}}} \right|, & \text{if } |w_{j,k}| \geq \lambda \\ 0, & \text{if } |w_{j,k}| < \lambda \end{cases},$$

$$\lambda = \sigma \sqrt{2 \ln N},$$

where  $w_{j,k}$  is the mixed data coefficient,  $\lambda$  is the noise threshold,  $N$  is the length of the data,  $\alpha$  and  $\beta$  are the adjustment factors, and  $\sigma$  represents the standard deviation of the noise in the data.

This function is continuous at the threshold point, overcoming the defect of non-differentiability of hard threshold functions at the threshold, and satisfying the odd function condition, which can maintain symmetry in the positive and negative parts of the processed data. In practical filtering processing,  $\alpha$  and  $\beta$  can be dynamically adjusted according to the size of the deviation and the actual situation to achieve better filtering results. Based on this, we propose a new model that uses signal-to-noise ratio as the fitness function and NOA to optimize WT parameters, quickly and accurately processing wind speed time series. The optimization process is shown in Figure 2.



**Figure 2.** The flowchart for NOA-WT parameters.

## 2.2. NOA-QRBiTCN-BiGRU model

### 2.2.1. Bidirectional temporal convolutional networks

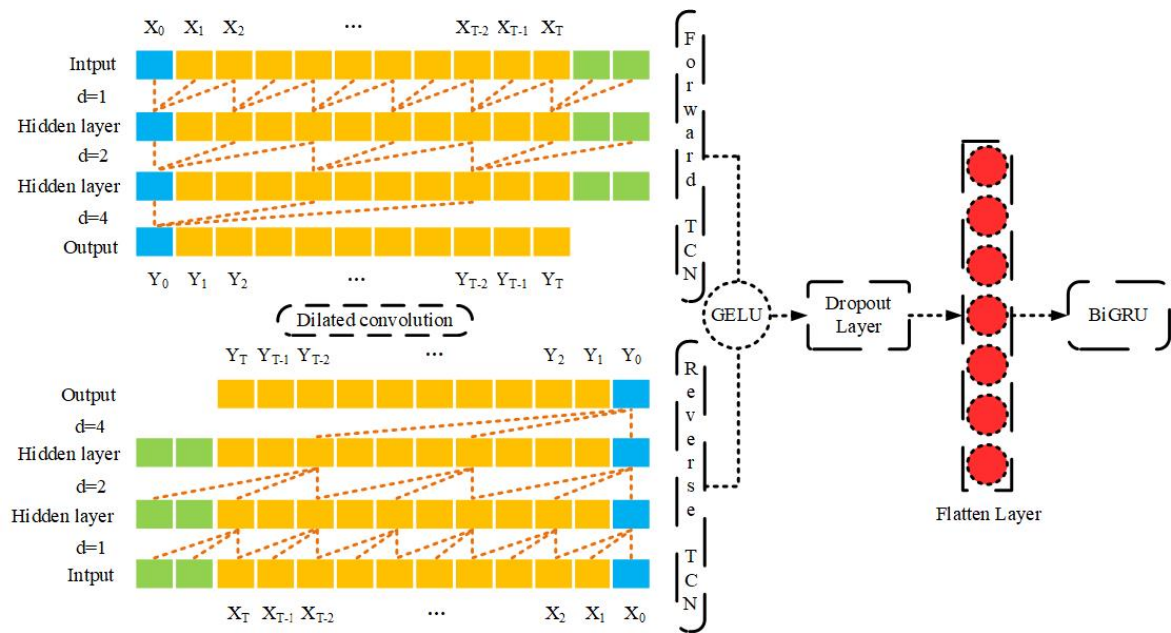
Bidirectional temporal convolutional networks (BiTCN) utilize convolutional neural networks (CNN) to efficiently process time series data. By combining past and future time information, they better capture the long-term dependencies between variables and improve prediction accuracy.

The module is shown below and the structure is shown in Figure 3.

(i) Expansion convolution: Stacking many diluted kernels by expanding the cardinality, skipping a certain number of input data points in the convolution operation, so that the convolution kernel can expand the receptive field without increasing computational cost.

(ii) GELU activation function: Using Gaussian error linear units instead of traditional ReLU activation functions, which allows the model to return some small negative values, avoiding the problem of neuron “death” and improving the learning ability of the model.

(iii) Dropout layer: By randomly discarding the outputs of some neurons, it prevents overfitting of the network and improves the generalization ability of the model.



**Figure 3.** BiTCN model structure.

### 2.2.2. Bidirectional gated recurrent unit network

Gated recurrent unit (GRU) is a recurrent neural network architecture specifically designed for processing sequential data, which introduces gating mechanisms to control information flow and memory updates. Unlike LSTM, GRU combines the forget gate and input gate into one update gate, simplifying the computation and structure of the network. The model structure is shown in Figure 4.

The calculation process of GRU unit is as follows:

1) Update gate: Determine how much of the current time step's state is influenced by past information and how much by the current input, allowing the network to selectively retain variable information.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z).$$

2) Reset gate: Determine how the current input is combined with past information. If the value is close to 0, it means that the network will discard the hidden state of the past and only use the current input for updates.

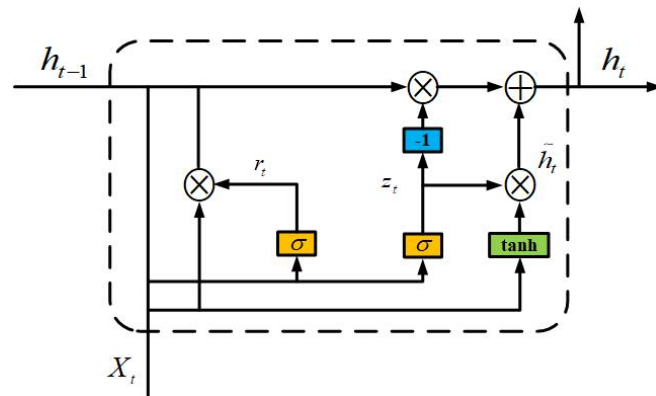
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r).$$

3) Hidden state: Adjust the past state based on the reset gate and, under the control of the update gate, combine the new input with the past hidden state to generate a new hidden state.

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h),$$

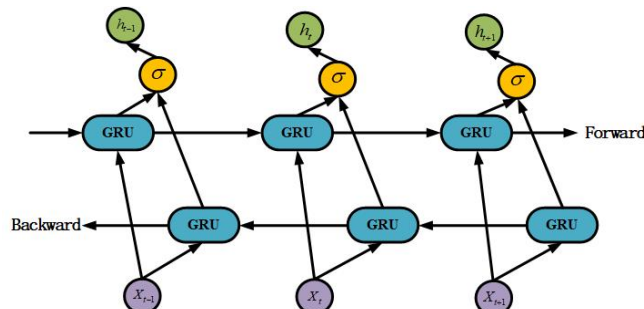
$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t,$$

where  $x_t$  represents the input data of the current time step,  $\sigma$  represents the sigmoid activation function,  $W$  represents the weight matrix of the update gate,  $h_{t-1}$  represents the hidden state of the previous time step, and  $b$  represents the bias term.



**Figure 4.** GRU model structure.

In GRU, hidden states only consider information from previous time steps and the current input. BiGRU uses two independent GRU networks for forward and backward processing of time series, respectively. The hidden state of each time step is a combination of two calculation results, so that the hidden state of each time step contains both previous historical information and future information. The model structure is shown in Figure 5.



**Figure 5.** BiGRU model structure.

### 2.2.3. NOA of BiTCN-BiGRU model parameters

This article takes wind speed interval prediction as an example to construct a model that accurately predicts wind speed with a 95% confidence interval. BiTCN has strong ability to capture local features and efficiently process local patterns in sequential data, while BiGRU performs well in learning global time series dependencies. This combination can better balance local and global information. In addition, BiTCN-BiGRU has fewer parameters and higher computational efficiency compared to other deep learning architectures such as the Transformer, which can reduce inference time while ensuring performance, as shown in the model comparison section. So, this paper chooses the BiTCN-BiGRU model for parameter optimization.

The main improvement is based on the BiTCN-BiGRU model. NOA is used to optimize the number of filters in the BiTCN model, the number of neurons in the BiGRU unit, and the learning rate and regularization parameters in the combination model, in order to find suitable combinations of hyperparameters in the complex search space. BiTCN can effectively process long time series data through convolution operations, capturing long-range temporal dependencies with fewer layers. This



approach facilitates gradient propagation in deep networks without the vanishing gradient problem seen in RNNs, making the model easier to train. Compared to LSTM, BiGRU has a simpler gating mechanism, resulting in lower computational complexity and faster training speed. The prediction process of NOA-BiTcn-BiGRU model is shown in Figure 6.

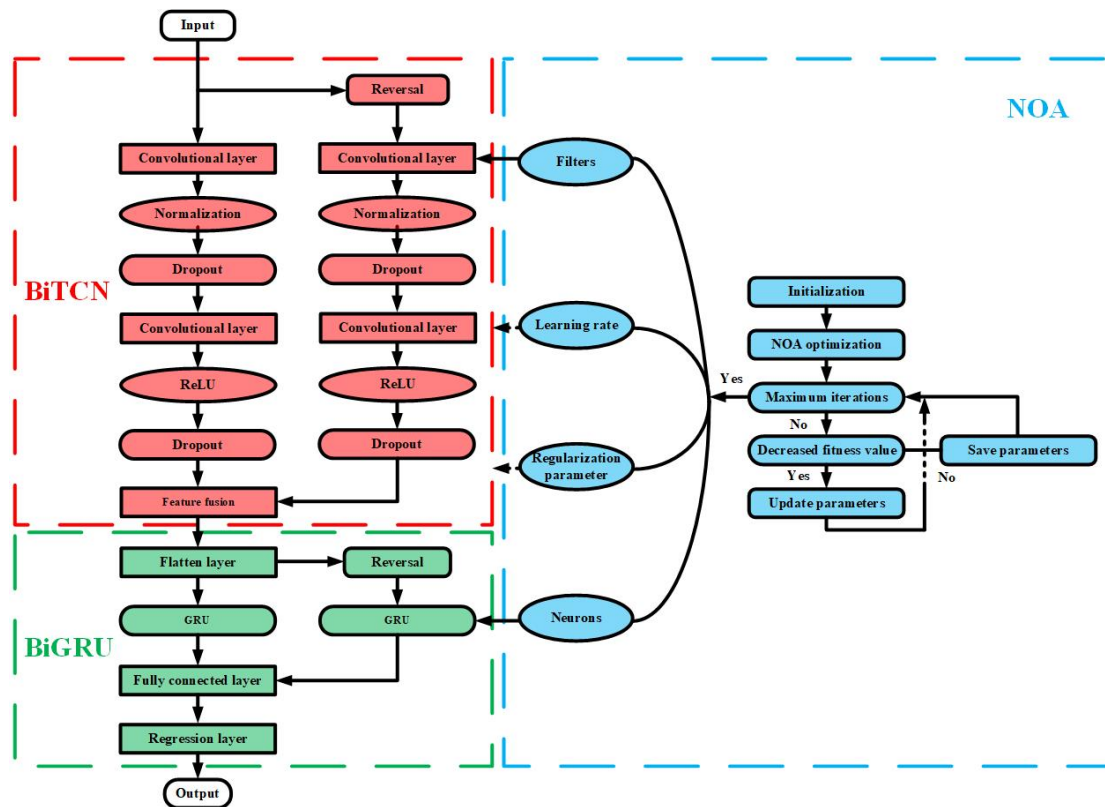


Figure 6. NOA-BiTcn-BiGRU model structure.

#### 2.2.4. Quantile regression

Quantile regression is a statistical method used to estimate the relationship between the conditional quantile of the dependent variable and the independent variable. Its goal is to predict a specific quantile of the dependent variable rather than predicting the mean or median. This method not only provides more comprehensive information about the data distribution but also serves as an effective tool for uncertainty prediction. By estimating multiple quantiles, quantile regression can construct prediction intervals, which explicitly capture the range of potential outcomes and quantify the uncertainty associated with predictions. This is particularly beneficial for regression tasks involving skewed or heteroscedastic data, where the variability of predictions cannot be fully captured by traditional methods. Therefore, quantile regression offers a robust framework for uncertainty prediction by accounting for the inherent randomness and variability in the data.

Quantile regression estimates regression coefficients at different quantiles by minimizing the following asymmetric loss functions:

$$L(\theta) = \min_{\xi \in CR} \left\{ \sum_{i: Y_i \geq \xi} \tau |Y_i - \xi| + \sum_{i: Y_i < \xi} (1 - \tau) |Y_i - \xi| \right\}. \quad (2.1)$$

Equation (2.1) can be equivalent to:

$$L(\theta) = \min_{\xi \in \mathbb{R}} \sum_{i=1}^n \rho_{\tau}(Y_i - \xi), \quad (2.2)$$

where  $\rho_{\tau}(u) = u(\tau - I(u < 0))$ .

### 2.3. Normality test

Kolmogorov-Smirnov (KS) test is a non parametric statistical method used to compare the differences between sample distributions and theoretical distributions, or to compare the differences between two sample distributions. The basic principle is to compare the maximum difference between the empirical distribution function  $F_n(x)$  and the theoretical distribution function  $F(x)$  of the sample. For normality testing, the theoretical distribution  $F(x)$  is the cumulative distribution function of the normal distribution, and its null and alternative hypotheses are:

$H_0$ : The sample comes from a normal distribution.

$H_1$ : The sample does not come from a normal distribution.

The empirical distribution function  $F_n(x)$  is a distribution function constructed based on sample data, used to approximate the true distribution. Its function definition is:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(X_i \leq x),$$

where  $I(X_i \leq x)$  is the indicator function.

In normality testing, the theoretical distribution function  $F(x)$  is the cumulative distribution function of the standard normal distribution, defined as:

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt.$$

The statistical definition of the KS test is:

$$D_n = \sup_x |F_n(x) - F(x)|,$$

where  $D_n$  represents the maximum difference between the empirical distribution function and the theoretical distribution function across all  $x$ , and  $\sup$  represents the maximum value. If  $D_n > D_{a,n}$  ( $D_{a,n}$  is the critical value corresponding to the significance level  $a$ ), the null hypothesis is rejected, and the sample is considered to not follow a normal distribution. If  $D_n < D_{a,n}$ , then the null hypothesis is accepted, and the sample is considered to follow a normal distribution.

### 2.4. T-test

The t-test is a statistical method used to compare the means of one or two samples to determine if the differences between them are statistically significant. It is based on the assumption that the data are approximately normally distributed and uses sample data to infer properties about the population. The primary goal is to assess whether the differences are caused by randomness or a true effect.

The t-test operates under the framework of hypothesis testing, which includes the following steps:

$H_0$ : Assumes no significant differences between the sample means.

$H_1$ : The sample does not come from a normal distribution.

The evaluation indicators for the model system proposed in this article need to be compared with other models one by one, so the two independent samples t-test is chosen. The formula for the t-statistic is as follows:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_p^2 \left( \frac{1}{n_1} + \frac{1}{n_2} \right)}}$$

where  $s_p^2$  is the pooled variance:

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2},$$

where  $\bar{x}_1$  and  $\bar{x}_2$  are the means,  $s_1$  and  $s_2$  are the standard deviations, and  $n_1$  and  $n_2$  are the sizes of the two samples. The p-value is calculated by comparing the t-statistic to the t-distribution for the corresponding degrees of freedom. If  $p \leq \alpha$ , the null hypothesis is rejected, indicating a statistically significant difference. If  $p > \alpha$ , the null hypothesis cannot be rejected, suggesting the difference is not statistically significant. While the data used in this paper does not fully follow a normal distribution, the central limit theorem provides a theoretical foundation for applying the t-test. The theorem states that when the sample size is sufficiently large, the sample mean tends to follow a normal distribution, regardless of the original data distribution. Therefore, the t-test is suitable for the analysis conducted in this paper.

## 2.5. Prediction and evaluation model

In order to evaluate the prediction results of the model, the evaluation metrics selected in this article include: mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), normalized cross correlation (NCC), prediction interval coverage probability (PICP) and prediction interval mean width (PIMW). The formulas are as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.3)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (2.4)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.5)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100, \quad (2.6)$$

$$\text{NCC}(u, v) = \frac{\sum_{x,y} (T(x, y) - \bar{T})(I(x + u, y + v) - \bar{I}_{u,v})}{\sqrt{\sum_{x,y} (T(x, y) - \bar{T})^2 \sum_{x,y} (I(x + u, y + v) - \bar{I}_{u,v})^2}}, \quad (2.7)$$

$$\text{PICP} = \frac{1}{n} \sum_{i=1}^n I(\cdot), \quad (2.8)$$

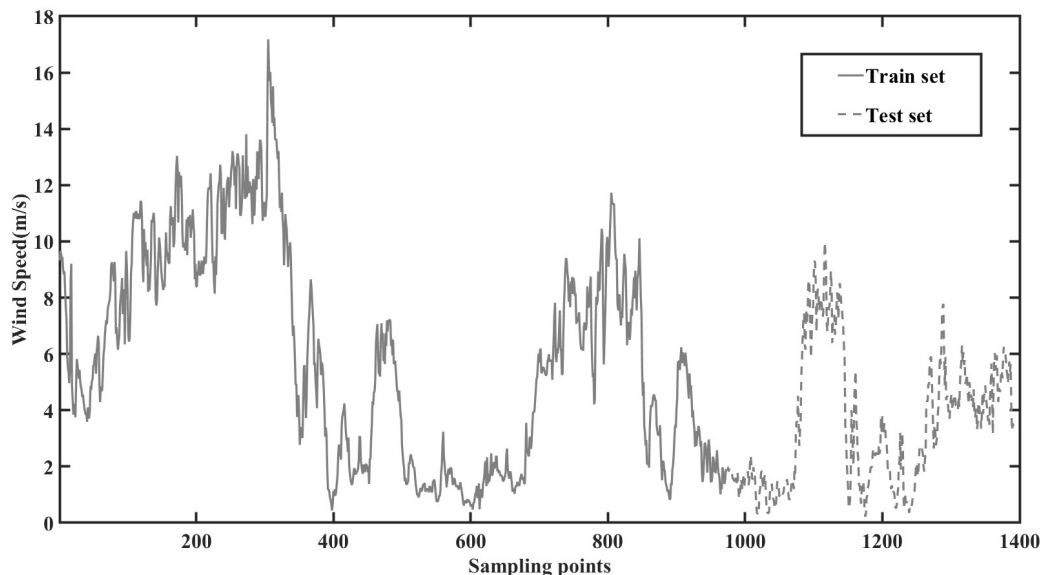
$$\text{PIMW} = \frac{1}{n} \sum_{i=1}^n (U_i - L_i), \quad (2.9)$$

where  $y_i$  represents the true value,  $\hat{y}_i$  represents the predicted value,  $n$  is the number of samples,  $I(\cdot)$  is the demonstrative function, and  $L_i$  and  $U_i$  are the lower and upper bounds of the prediction interval, respectively.  $T(x, y)$  corresponds to a data point in the time series,  $I(x + u, y + v)$  represents a sliding window of the time series,  $\bar{T}$  represents the average value of the data, and  $\bar{I}_{u,v}$  represents the average value of the sliding window.

MSE reflects the square mean of the error between predicted and actual values, RMSE reflects the degree of deviation of predicted values from actual values, MAE reflects the average absolute value of the error between predicted and actual values, and MAPE reflects the relative magnitude of prediction errors. The smaller the value, the higher the prediction accuracy. NCC reflects the similarity between two time series, with values closer to 1 indicating a better match. PICP reflects whether the confidence level of the prediction interval is high enough, while PIMW reflects the accuracy of the prediction interval. Overly wide prediction intervals may lead to coverage approaching the target, but the accuracy of the prediction is poor. A too narrow prediction interval may improve accuracy but leads to insufficient coverage, so a balance needs to be found between PICP and PIMW.

### 3. Example analysis

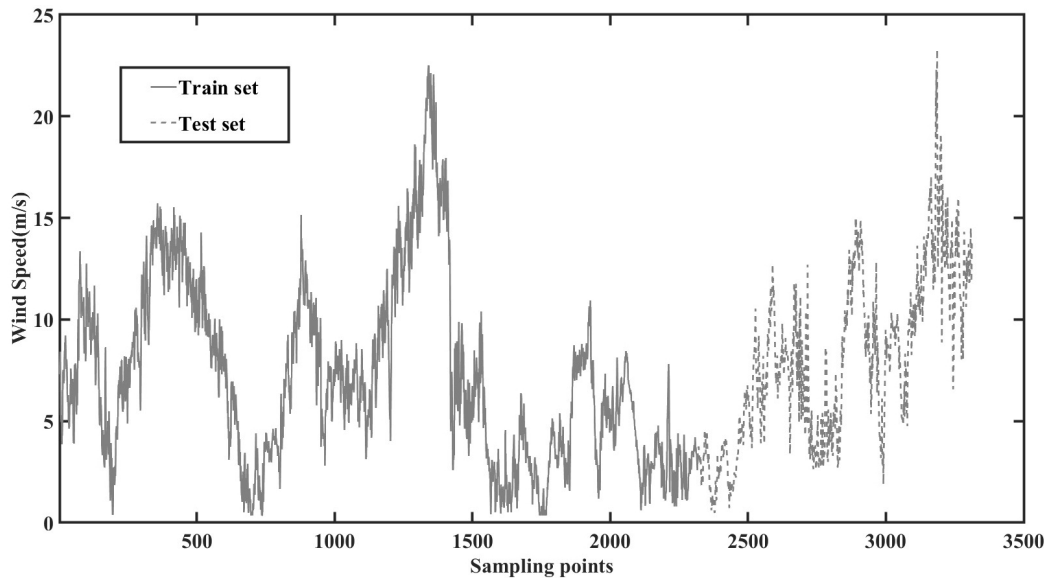
#### 3.1. Data sources



**Figure 7.** Data1 wind speed dataset.

We used two datasets to predict the nonlinear time series analysis. The first dataset (data1) is from the public dataset of the 2022 Baidu KDD CUP competition [6], with a time resolution of 10 minutes.

A total of 1391 data points were selected, as shown in Figure 7. The second dataset (data2) is sourced from the Sotavento Galicia wind farm in Spain ([www.sotaventogalicia.com](http://www.sotaventogalicia.com)), with a time resolution of 10 minutes. A total of 3312 data points were selected, as shown in Figure 8. After verification, there were no missing values in the selected data. The first part of the dataset was used as training data, and the second part as testing data. MATLAB was used to process and analyze it.



**Figure 8.** Data2 wind speed dataset.

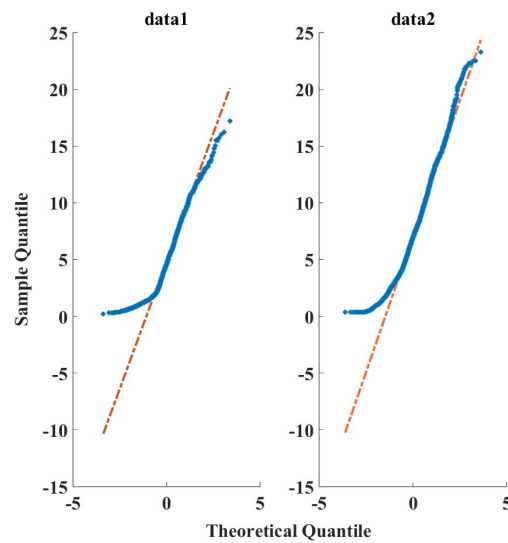
### 3.2. Normality test

Before starting the data analysis, a KS test was conducted on data1 and data2, and the analysis results are shown in Table 2 and Figure 9.

**Table 2.** KS test result.

	P-value	$D_n$	Skewness	Kurtosis
data1	5.5014e-15	0.1095	0.5920	2.3683
data2	1.5698e-11	0.0620	0.6218	2.9362

From the above, it can be seen that the p-value of dataset data1 is 5.5014e-15, which is far below the significance level ( $\alpha = 0.05$ ), indicating a significant difference between the dataset and the normal distribution.  $D_n$  shows that the maximum difference between the data and the normal distribution is 0.1095, indicating sufficient evidence to reject the hypothesis that the data follows a normal distribution. The p-value of data2 is 1.5698e-11, which is much lower than the significance level ( $\alpha = 0.05$ ), indicating that the dataset deviates significantly from a normal distribution.  $D_n$  indicates that the maximum difference between the data and the normal distribution is 0.0620; therefore, the data does not conform to the normality assumption. The skewness of the two datasets is close to 0.6, indicating that there are large values in the datasets and both have a right-skewed trend. Both of these kurtosis values are slightly lower than 3, indicating a relatively flat distribution with lower peaks and lighter tails. The data is evenly distributed in the central region with fewer extreme values at the tails.



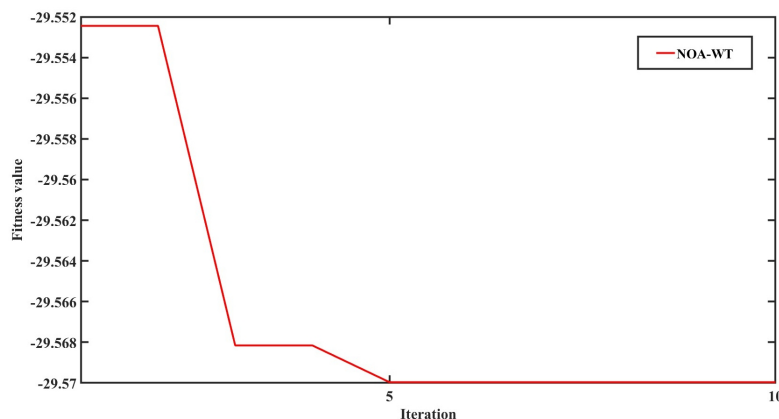
**Figure 9.** Data QQ chart.

Based on the above analysis, QR is very suitable for processing skewed data, estimating the relationship between the dependent variable and the independent variable at different quantiles, effectively capturing the changing characteristics of the data in different parts, such as the left tail, median, and right tail, which will better handle the complex distribution relationships in skewed data.

### 3.3. Example analysis

#### 3.3.1. Wind speed interval prediction based on data1

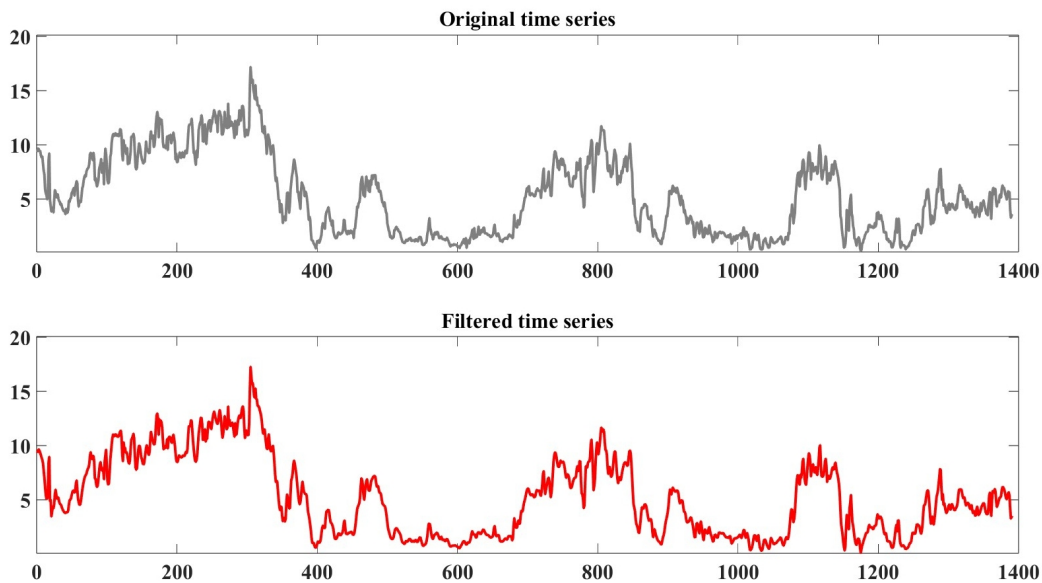
Using the data1, when initializing the NOA, the population size was set to 30, the maximum number of iterations to 10, and the range of adjustment factors  $\alpha$  and  $\beta$  to  $[1, 10]$  and  $[1, 10]$ . At the fifth iteration, the fitness value was 29.57 and the curve tended to flatten, as shown in Figure 10.



**Figure 10.** NOA optimization curve for data1.

The optimal adjustment factors  $\alpha = 10$  and  $\beta = 0.1$  were introduced into the WT model. The wavelet basis was set to db5, and the wavelet decomposition level was set to 1 for filtering the wind speed time series. The NCC of 0.9994 indicates a very high similarity between the two time series,

which means that the filtering process preserved as much useful information as possible. The results are shown in Figure 11.



**Figure 11.** Comparison of filtered time series for data1.

Using the C-C algorithm [32], the time delay  $\tau = 32$  and embedding dimension  $d = 3$  of the wind speed time series were determined. The Wolf method [33] was then applied to assess whether the reconstructed wind speed time series exhibits chaotic characteristics. The Lyapunov exponent was calculated to be 0.1671, indicating that the time series is chaotic. Subsequently, the NOA-QR-BiTCN-BiGRU model was employed for interval prediction.

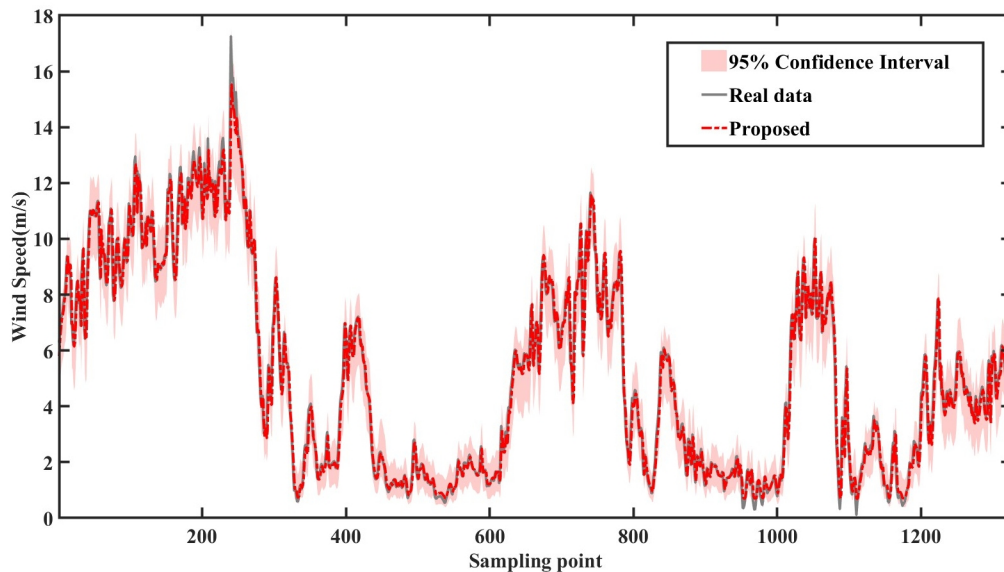
**Table 3.** Hyperparameter and network structure settings.

Model&Algorithm	Hyperparameter	Value
NOA & other comparison algorithms	Population	5
	Iteration	5
	Convolutional kernel	$5 \times 5$
BiTCN-BiGRU & other comparison models	Dropout rate	0.01
	Block	1
	Initial learning rate	0.001
	Filters	8
	Neurons	13
	Gradient descent rate	Adam
	Gradient threshold	0.9
	Training epochs	30
	Learning rate decay factor	0.1
	Learning rate decay	10

The NOA was used to optimize the parameters of the BiTCN-BiGRU model to obtain reasonable parameter values. The parameters optimized included the number of filters, the number of neurons,

the initial learning rate, and the regularization parameter. The upper bound of the parameter range was [0.1, 18, 10, 0.01] and the lower bound was [0.001, 3, 1, 0.00001]. The remaining hyperparameters and network structure settings are shown in Table 3.

In the optimization process, the RMSE was used as the objective function. Upon completion of the iterations, the parameter values corresponding to the minimum fitness value were saved as the optimized parameters. The proposed model was then applied to perform interval prediction on the filtered signal after phase space reconstruction. The predicted values closely match the actual values, as shown in Figure 12.



**Figure 12.** Wind speed interval prediction for data1.

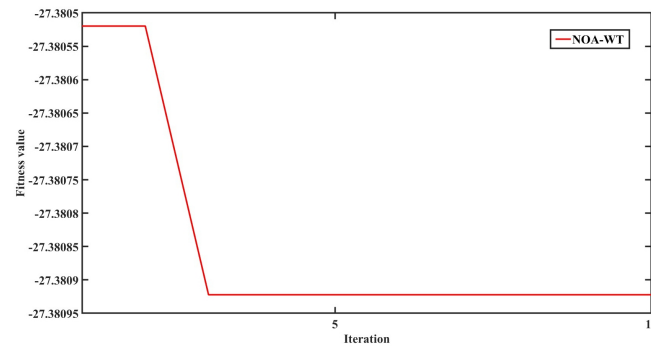
The NOA-QRBiTcN-BiGRU model achieved the smallest prediction error, with MSE, RMSE, and MAE values of 0.2938, 0.5421, and 0.3799, respectively, indicating the highest prediction accuracy. The PICP value was 0.9607, showing that the predicted interval of this model covers approximately 96.07% of the actual values. This coverage rate is higher than that of other models, making it more capable of capturing the real data, thereby demonstrating high reliability. The PIMW value was 64.0593, indicating that the prediction interval of this model is relatively narrow, which means that the model provides accurate predictions with minimal redundancy in the prediction range. Therefore, the NOA-QR-BiTcN-BiGRU model not only has strong interval coverage ability but also achieves high accuracy with a smaller prediction interval.

### 3.3.2. Wind speed interval prediction based on data2

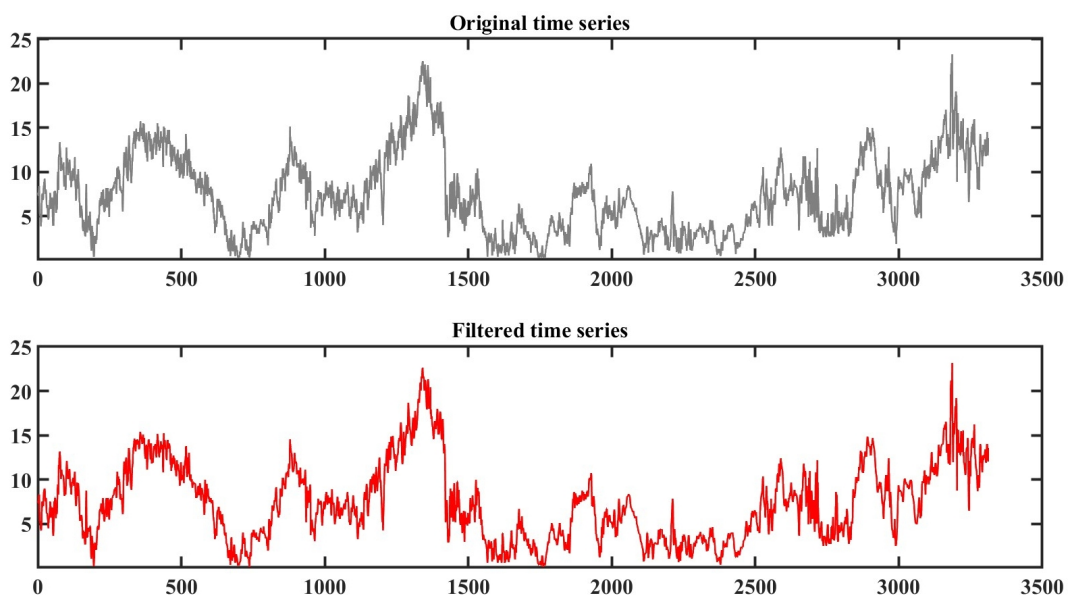
Using the data2 dataset, the NOA was initialized with identical parameters. At the third iteration, the fitness value was -27.38092, and the curve began to flatten, as shown in Figure 13.

The optimal adjustment factors  $\alpha = 10$  and  $\beta = 100$  were applied to the WT model with the same parameters to filter the wind speed time series. The NCC was 0.9990, indicating a very high similarity between the two datasets, meaning that useful information was preserved as much as possible during the filtering process. The results are shown in Figure 14.





**Figure 13.** NOA curve for data2.

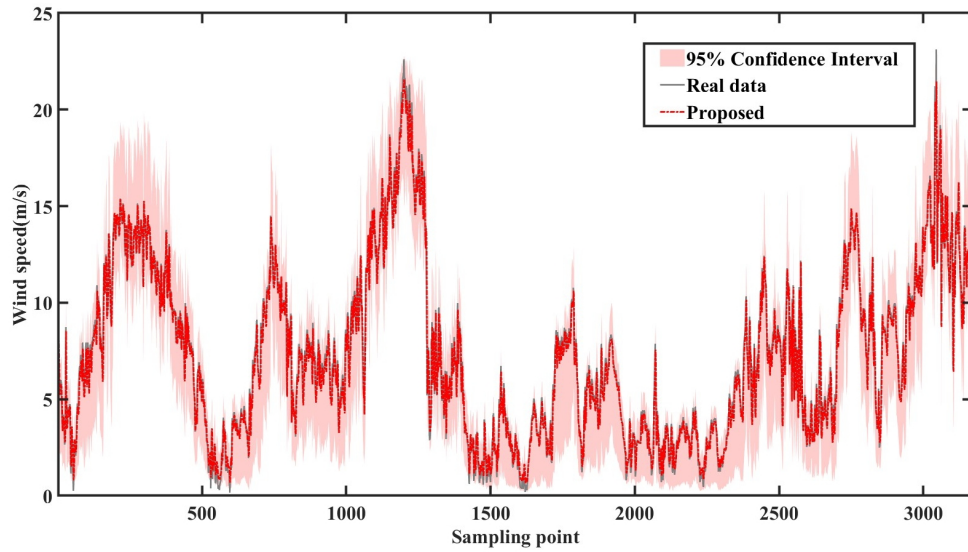


**Figure 14.** NOA-WT filtering results for data2.

Using the C-C algorithm, the time delay  $\tau = 35$  and embedding dimension  $d = 4$  of the wind speed time series were determined. The Wolf method was then applied to assess whether the reconstructed wind speed time series exhibits chaotic characteristics. The calculated Lyapunov exponent was 0.1862, indicating that the time series possesses strong nonlinearity. Subsequently, the NOA-QR-BiTCN-BiGRU model was used for interval prediction. Then, using the same hyperparameters and network structure, the proposed model was applied to perform interval prediction on the filtered data after phase space reconstruction. The predicted values closely matched the actual values, as shown in Figure 15.

NOA-QRBiTCN-BiGRU had the smallest prediction error, with MSE, RMSE, and MAE of only 0.6391, 0.7994, and 0.5665, indicating that the model has high prediction accuracy and can effectively capture trends and features in the data. The PICP was 0.9719, indicating that the predicted interval of the model can cover approximately 97.19% of the true values, with high interval reliability and providing robust interval prediction results. PIMW was 93.0, and the prediction interval of this model was the narrowest among all compared models, indicating that while ensuring high coverage, the model can effectively narrow the prediction interval, improving the accuracy and efficiency of prediction. Therefore, NOA-QRBiTCN-BiGRU not only has excellent point prediction performance, but

also demonstrates good balance in interval prediction, achieving a good balance between coverage and interval width.

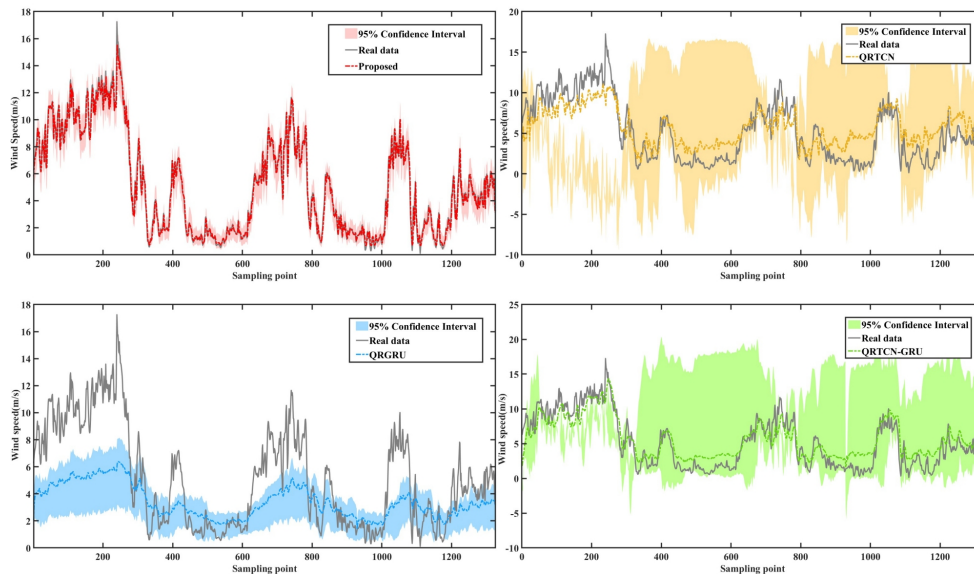


**Figure 15.** Wind speed interval prediction for data2.

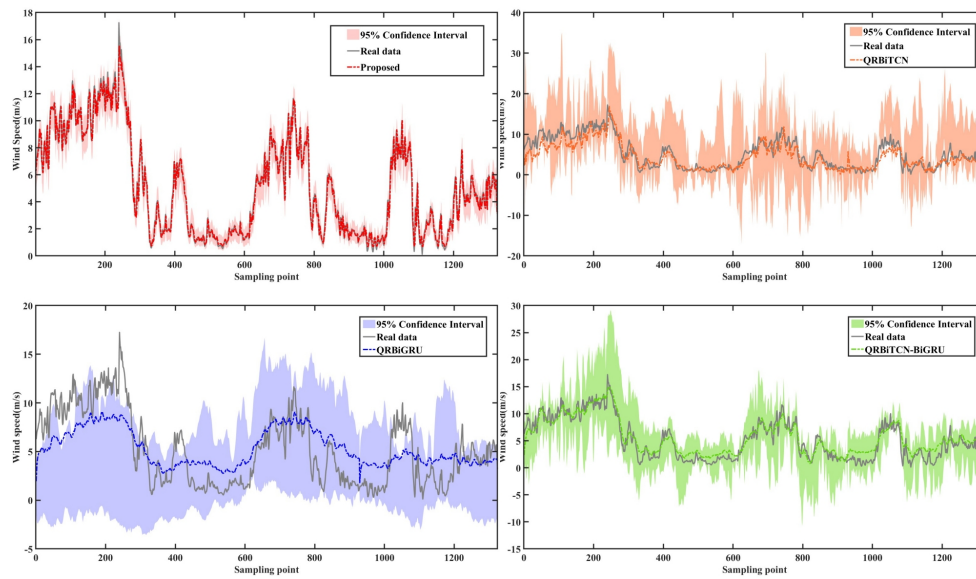
### 3.4. Experimentation comparison

#### 3.4.1. Result analysis based on data1

In all single machine learning models and combined deep learning models, the same hyperparameters and network structure were used to study the impact of different model combination methods on prediction accuracy. The results are shown in Table 4 and Figures 16 and 17.



**Figure 16.** Unidirectional deep learning model prediction for data1.



**Figure 17.** Bidirectional deep learning model prediction for data 1.

**Table 4.** Model evaluation indicators for data 1.

Model	MSE	RMSE	MAE	PICP	PIMW
QRTCEN	4.8597	2.2045	1.8738	0.6847	593.7037
QRGRU	10.1017	3.1783	2.4091	0.4366	102.8994
QRTCEN-GRU	2.7573	1.6605	1.4028	0.7699	557.7269
QRBiTCN	3.0375	1.7428	1.3385	0.8914	497.9103
QRBiGRU	6.467	2.5430	2.1771	0.7458	358.8504
QRBiTCN-BiGRU	1.5125	1.2298	1.0106	0.9426	314.8444
Proposed	0.2938	0.5421	0.3799	0.9607	64.0593

From the above, the combined models (QRTCEN-GRU and QRBiTCN-BiGRU) achieved average reductions in MSE, RMSE, and MAE by 65.09%, 40.21%, and 38.10%, respectively, compared to single machine learning models (QRTCEN, QRGRU, QRBiTCN, and QRBiGRU). This indicates that the combined models can significantly reduce prediction errors. The PICP interval coverage rate increased by an average of 16.66%, showing that the combined models are more reliable when predicting uncertainty, with a higher likelihood of predicting that the true values fall within the intervals. Additionally, the PIMW interval width decreased by an average of 47.94%, demonstrating that the combined models not only improve coverage but also provide more compact and precise prediction intervals. Therefore, the combined models integrate advantages of multiple models, better capturing data complexity and improving prediction performance.

Compared with the unidirectional deep learning model, the MSE, RMSE, and MAE of the bidirectional deep learning model decreased by an average of 4.20%, 2.41%, and 2.26%, indicating that the bidirectional model can learn from both forward and backward time information and still bring a certain degree of error reduction. The average increase in PICP was 22.95%, indicating that the bidirectional model is slightly more reliable in the prediction interval and can better envelop the actual

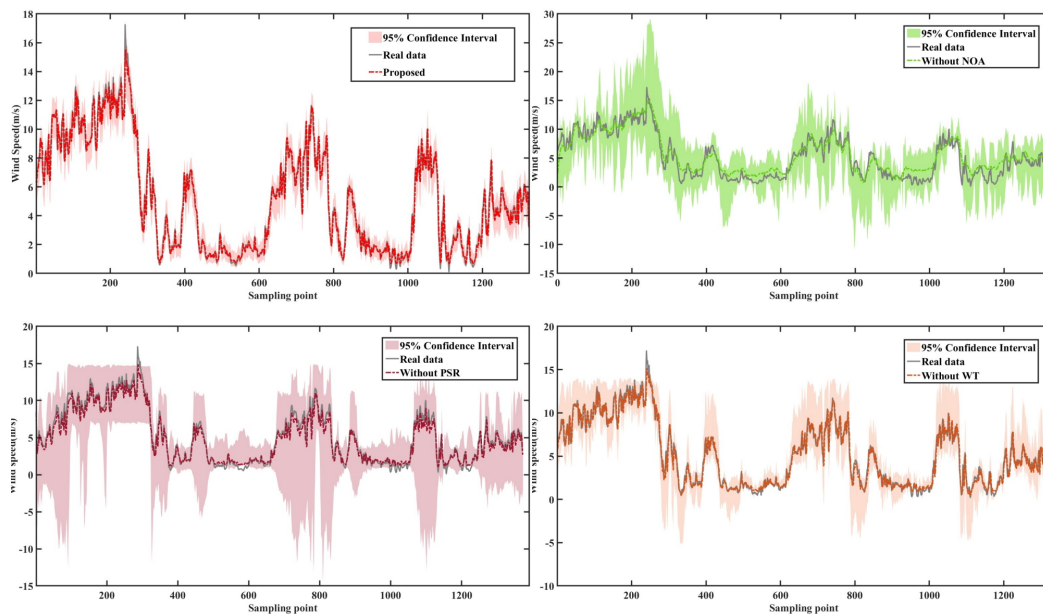
values. The average reduction of PIMW was 27.57%, indicating that the bidirectional deep learning model can maintain the compactness of the prediction interval while increasing coverage, reducing unnecessary interval width and making predictions more accurate.

### 3.4.2. Ablation experiment based on data1

The purpose of the ablation experiment is to evaluate the contribution of each component to the overall model performance by gradually removing components or modules from the model and observing the changes in performance. In the proposed model system, the NOA, PSR technique, and WT were removed one by one. Three comparative experiments were designed to understand the impact of each part on prediction accuracy, as shown in Table 5 and Figure 18.

**Table 5.** Evaluation indicators for ablation experiments for data1.

Method	MSE	RMSE	MAE	PICP	PIMW
Without NOA	1.5125	1.2298	1.0106	0.9426	314.8444
Without PSR	0.5240	0.7239	0.5537	0.9096	231.9515
Without WT	0.4054	0.6367	0.4583	0.9223	134.9661
Proposed	0.2938	0.5421	0.3799	0.9607	64.0593



**Figure 18.** Prediction of the ablation experiment for data1.

From the above, it is clear that the NOA has the largest impact on the model. It reduced the MSE, RMSE, and MAE errors by 80.57%, 55.91% and 62.40%, respectively, increased the PICP interval coverage rate by an average of 1.81%, and reduced the PIMW interval width by an average of 250.78%. This indicates that the prediction interval becomes significantly narrower, thereby improving prediction accuracy. The PSR technique reduced MSE, RMSE, and MAE by 43.92%, 25.11% and 31.37%, respectively, increased the PICP interval coverage rate by an average of 5.1%, and reduced the PIMW

interval width by 167.89%. This shows that it has a notable effect on improving the coverage rate. The WT technique had the smallest contribution to the model's performance, reducing MSE, RMSE, and MAE by 27.52%, 14.86% and 17.09%, respectively. It increased the PICP interval coverage rate by an average of 3.8% and decreased the PIMW interval width by 70.90%, showing that it can enhance the model's performance to a certain extent.

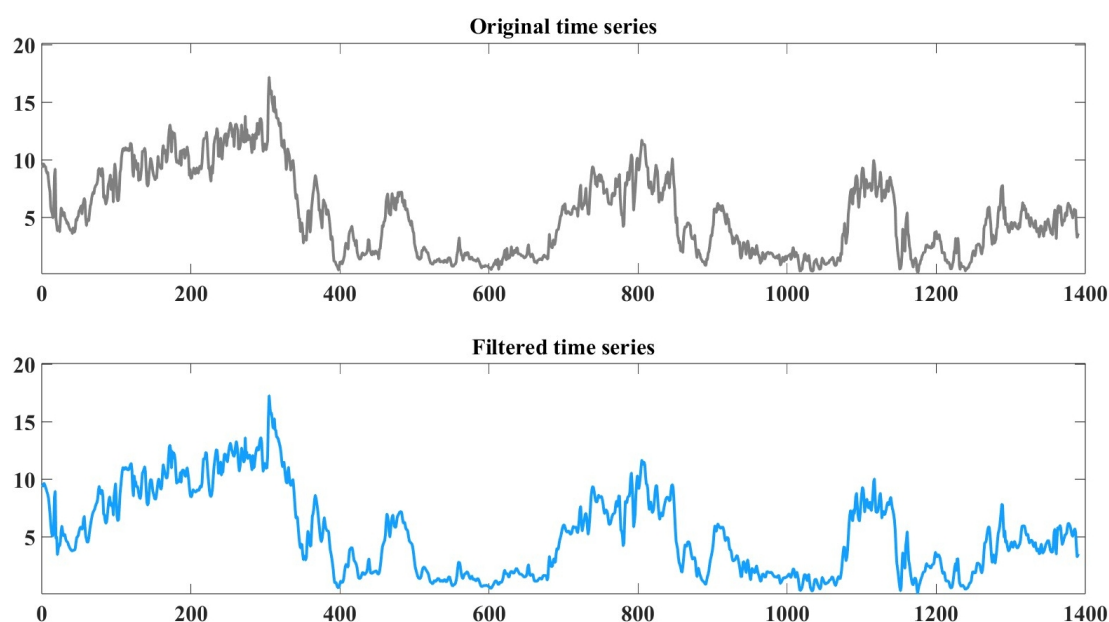
**Table 6.** Significance test of ablation experiments for data1.

$p$	MSE	RMSE	MAE	PICP	PIMW
Without NOA	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$	0.9225	$\ll 0.0001$
Without PSR	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$	0.1811	0.0021
Without WT	0.0003	0.0001	0.0003	0.0556	0.0072

From Table 6, NOA, PSR, and WT modules used in the proposed model system all passed the significance tests for all error metrics, indicating that these modules are indispensable for improving prediction accuracy. Additionally, the model also passed the significance test for PIMW, further demonstrating its ability to effectively reduce the prediction interval width. Although no significant differences were observed in the PICP, the proposed model system shows remarkable advantages in optimizing interval compactness and enhancing prediction accuracy.

### 3.4.3. Algorithm comparison based on data1

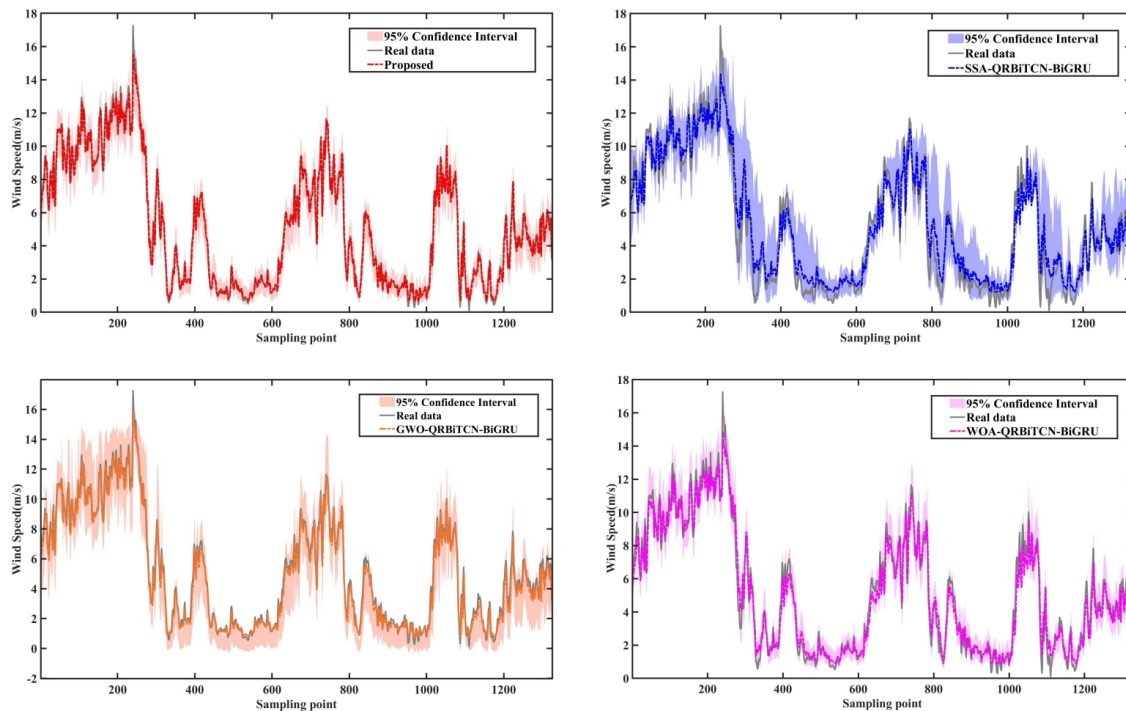
To demonstrate that NOA provides higher prediction accuracy in optimizing deep learning models, it was compared with state-of-the-art optimization algorithms, including SSA [34], GWO [35], and WOA [36]. These algorithms are widely used due to their excellent performance. For a fair comparison, an experiment was designed in which deep learning models were optimized using both the NOA and other algorithms, ensuring that the experiments were conducted under the same hyperparameters and network structure conditions to objectively assess the optimization effectiveness of each algorithm.



**Figure 19.** Filtering results for data1.

First, the optimization algorithms were used to optimize the adjustment factor  $\{\alpha, \beta\}$  for the WT filtering model, obtaining the same optimal parameter combination  $\{10, 0.1\}$  as the NOA-WT model, with a corresponding fitness value of  $-29.57$ . The optimal adjustment factors  $\alpha = 10$  and  $\beta = 0.1$  were then applied to the WT model to filter the wind speed time series. The filtering results are shown in Figure 19.

Next, the data filtered by the WT model optimized based on other algorithms were subjected to phase space reconstruction. The Lyapunov exponent was  $0.1671$ , indicating that the time series is chaotic. Therefore, after PSR, the QRBiTCN-BiGRU model optimized based on other algorithms was used for interval prediction. The final prediction results are shown in Figure 20.



**Figure 20.** Interval prediction for data1 by the NOA and other optimization algorithms.

As shown in Table 7, the model optimized with NOA reduced MSE, RMSE, and MAE by an average of  $37.29\%$ ,  $21.73\%$  and  $27.02\%$ , respectively, when compared with other algorithms, indicating smaller prediction errors. The PICP increased by an average of  $10.55\%$ , suggesting that the prediction interval covers a higher proportion of actual values, making it more reliable. Additionally, the PIMW decreased by an average of  $26.66\%$ , meaning that the prediction interval is more precise and concentrated, aligning more closely with the actual situation.

**Table 7.** Model evaluation indicators for data1.

Method	MSE	RMSE	MAE	PICP	PIMW
With SSA	0.7365	0.8582	0.6685	0.8868	131.067
With GWO	0.34283	0.58551	0.4278	0.74359	75.3167
With WOA	0.47017	0.68569	0.51857	0.93514	65.7983
With NOA	0.2938	0.5421	0.3799	0.9607	64.0593

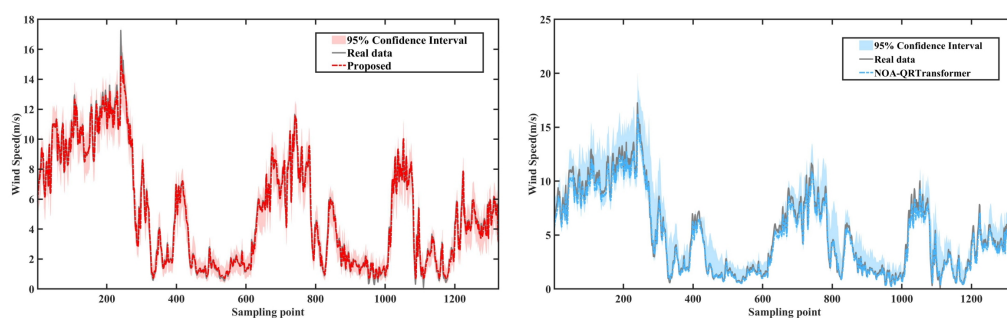
After conducting 10 independent repeated experiments and performing significance tests on the results, it can be seen from Table 8 that the model optimized by NOA achieved p-values lower than 0.001 for error metrics, indicating that its prediction accuracy is significantly better than other algorithms with strong statistical significance. For PICP, all algorithms had p-values greater than 0.05, showing no statistically significant differences. This suggests that the performance of the algorithms on interval coverage rate is comparable. However, for PIMW, the p-values for the model optimized by NOA were all lower than 0.05, passing the significance test. This indicates that, despite having the same interval coverage rate, the model optimized by NOA effectively reduces the prediction interval width, thus improving the compactness and precision of the interval prediction.

**Table 8.** Model evaluation indicators for data1.

$p$	MSE	RMSE	MAE	PICP	PIMW
With SSA	0.0008	0.0003	0.0002	0.363	0.0006
With GWO	0.0015	0.0011	0.0013	0.0938	0.0036
With WOA	0.0188	0.0157	0.0075	0.3482	0.0398

#### 3.4.4. Model comparison based on data1

In order to comprehensively verify the performance advantages of the proposed model, this section designed a comparative experiment with Transformer, aiming to highlight its high accuracy and faster inference speed in practical applications. Transformer, as one of the most popular model architectures in the field of deep learning, relies on its powerful feature extraction capabilities and wide applicability. In this experiment, only the prediction model was replaced with Transformer, while the other modules (such as NOA and PSR) maintained the same parameters and structural design, named as NOA-QRTransformer. The experimental results show that the proposed model, with its lightweight architecture and efficient design, significantly reduces inference time while achieving high-precision prediction, fully demonstrating its superiority in resource constrained scenarios, as shown in Figure 21.



**Figure 21.** Model comparison for data1.

From Table 9, it can be seen that MSE, RMSE, and MAE of the proposed model were 0.2938, 0.5421 and 0.3799, respectively, significantly lower than those of the NOA-QRTransformer. Additionally, the proposed model had a PICP of 0.9607 and a PIMW of 64.0593, which is much smaller than those of the NOA-QRTransformer. Therefore, the proposed model outperforms in all indicators, exhibiting higher accuracy and lower uncertainty.

**Table 9.** Model evaluation indicators for data1.

Method	MSE	RMSE	MAE	PICP	PIMW
Proposed	0.2938	0.5421	0.3799	0.9607	64.0593
NOA-QRTransformer	0.4584	0.6770	0.5043	0.8755	85.7895

In addition, the error index and PIMW of the proposed model passed the significance test, with all p-values lower than 0.05. This indicates that the proposed model not only has high prediction accuracy under the same PICP but also reduces the average prediction interval width, further verifying the statistical significance of the model performance, as shown in Table 10.

**Table 10.** Significance test of model comparison for data1.

$p$	MSE	RMSE	MAE	PICP	PIMW
NOA-QRTransformer	0.0003	0.0001	$\ll 0.0001$	0.0651	0.0001

In 10 experiments, the average inference time of the proposed model was 136 seconds, significantly lower than NOA-QRTransformer's 192 seconds, fully demonstrating its lightweight architecture. By combining shorter inference time with higher prediction accuracy, this model achieves a better balance between performance and resource consumption, demonstrating strong practicality.

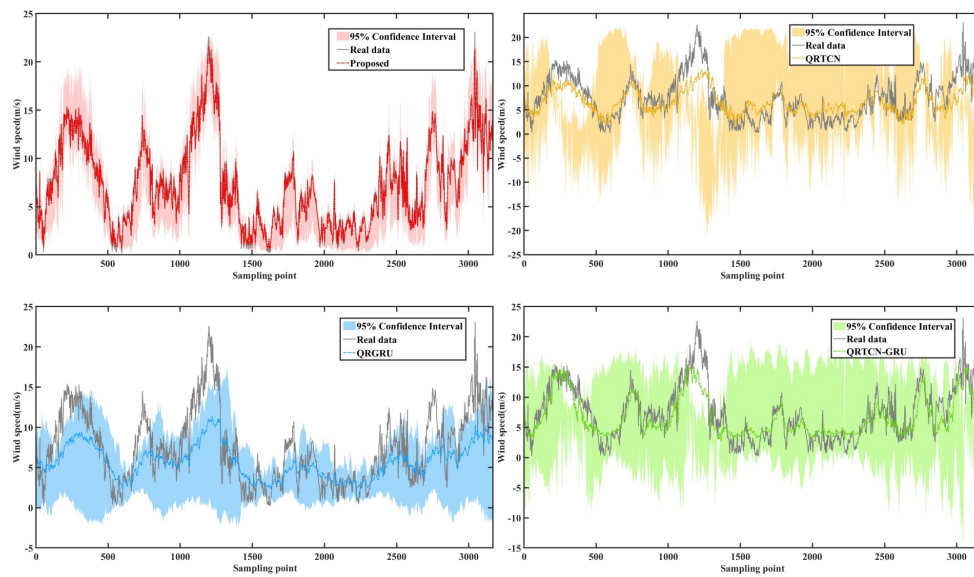
#### 3.4.5. Result analysis based on data2

The impact of different model combination methods on prediction accuracy, with identical hyper-parameters and network structures set across all machine learning models and deep learning models, was evaluated. The results are presented in Table 11 and Figures 22 and 23.

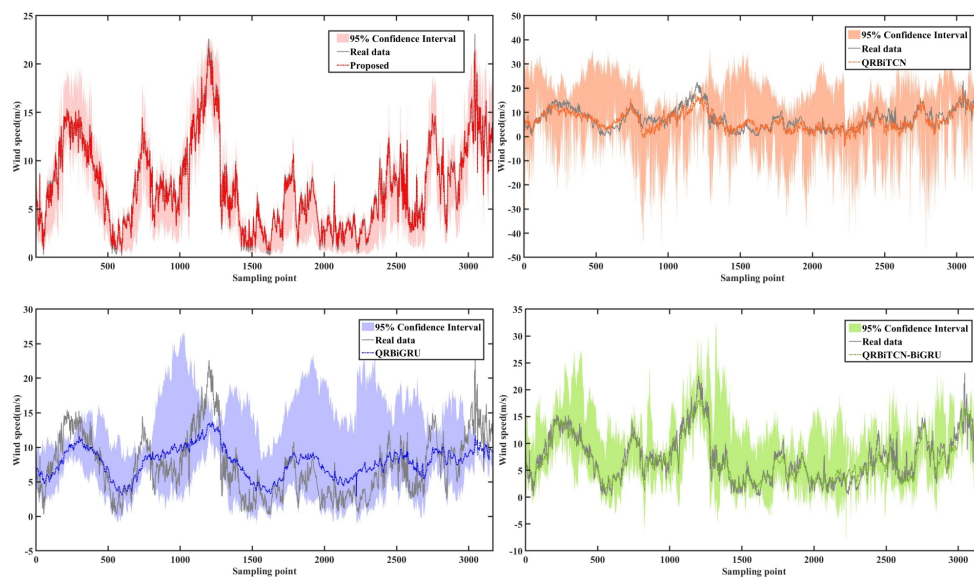
**Table 11.** Model evaluation indicators for data2.

Model	MSE	RMSE	MAE	PICP	PIMW
QRTCEN	8.7008	2.9497	2.3581	0.6612	520.9449
QRGRU	10.7522	3.2790	2.5025	0.6720	169.3105
QRTCEN-GRU	5.7098	2.3895	1.8592	0.7906	385.1367
QRBiTCN	7.0178	2.6491	2.1169	0.8432	714.9137
QRBiGRU	9.1912	3.0317	2.5309	0.8275	307.8303
QRBiTCN-BiGRU	2.2942	1.5147	1.1887	0.9113	267.7013
Proposed	0.6391	0.7994	0.5665	0.9719	93.0800





**Figure 22.** Unidirectional deep learning model prediction for data2.



**Figure 23.** Bidirectional deep learning model prediction for data2.

From the above, it can be observed that the ensemble models (QRTCN-GRU and QRBiTCN-BiGRU) show an average reduction in MSE, RMSE, and MAE by 55.11%, 34.43% and 35.89%, respectively, compared to the machine learning data models (QRTCN, QRGRU, QRBiTCN, and QRBiGRU). This indicates that the ensemble models significantly improve prediction accuracy, better fitting the data and reducing prediction errors. The PICP increased by an average of 9.99%, suggesting that the ensemble models provide higher reliability in interval predictions, more comprehensively covering the actual values. Additionally, the PIMW decreased by an average of 101.83%, indicating that the ensemble models successfully narrowed the prediction interval while maintaining high coverage, thereby improving the efficiency and precision of interval predictions.

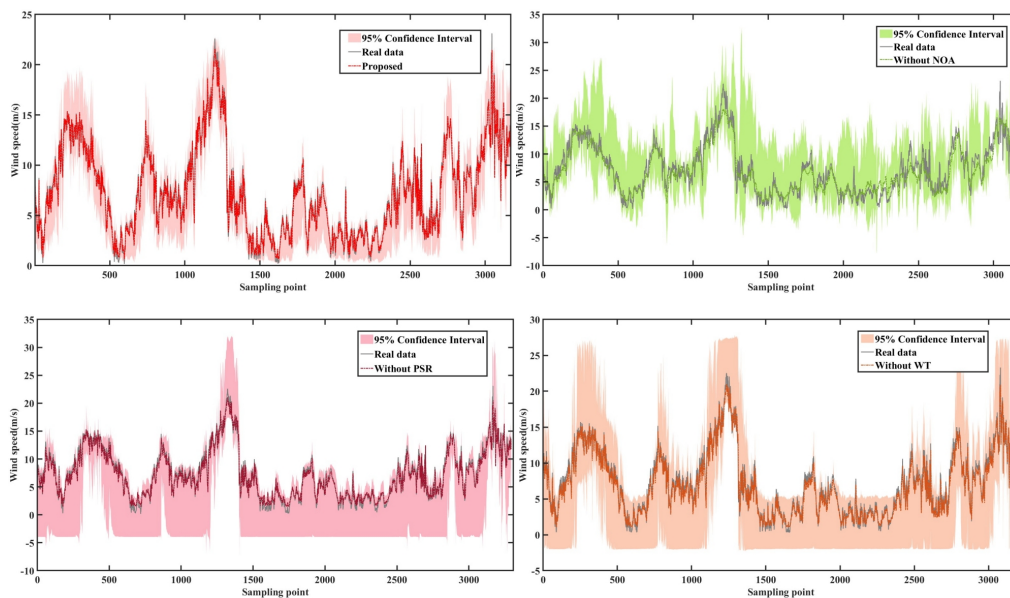
Compared to unidirectional deep learning models, the bidirectional deep learning models show an average reduction in MSE, RMSE, and MAE, by 26.46%, 16.50% and 13.14%, respectively, indicating that the bidirectional models significantly improve prediction accuracy and are better able to comprehensively capture data features and trends. The PICP increased by an average of 15.27%, suggesting that bidirectional models are more reliable in interval predictions, covering more actual values. However, the PIMW increased by an average of 71.68%, indicating that the prediction intervals of bidirectional deep learning models are wider, which may lead to interval redundancy. This suggests that there is still room for improvement in optimizing interval width. Therefore, while bidirectional models enhance accuracy and coverage, further optimization of interval width is needed to improve overall efficiency.

#### 3.4.6. Ablation experiment based on data2

Keeping the same parameter settings and network structure design, three comparative experiments were designed by removing the NOA, PSR technique, or WT technique, to understand the impact of each component on prediction accuracy. The results are shown in Table 12 and Figure 24.

**Table 12.** Evaluation indicators for ablation experiments for data2.

Method	MSE	RMSE	MAE	PICP	PIMW
Without NOA	2.2942	1.5147	1.1887	0.9113	267.7013
Without PSR	0.7234	0.8505	0.6316	0.9459	231.3922
Without WT	1.0832	1.0408	0.7817	0.9298	228.1087
Proposed	0.6391	0.7994	0.5665	0.9719	93.08



**Figure 24.** Prediction of ablation experiment for data2.

From the above, it can be seen that NOA had the greatest impact on the model, reducing MSE, RMSE, and MAE by 72.14%, 47.21% and 52.34%, respectively. PICP increased by an average of

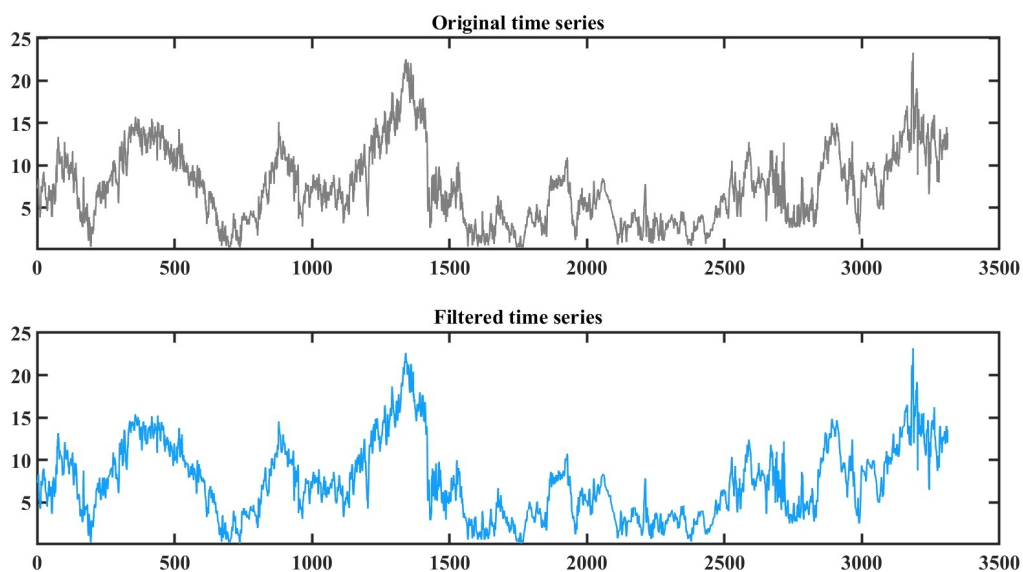
6.05%, and PIMW decreased by an average of 174.62%, indicating that the prediction intervals have significantly narrowed while coverage has improved, greatly enhancing the model's prediction accuracy. Second, the WT technique reduced MSE, RMSE, and MAE by 40.99%, 23.18% and 27.52%, respectively, with the PICP increasing by an average of 4.21%, and PIMW decreasing by an average of 135.02%. This shows a notable effect in reducing errors and improving interval coverage, thereby enhancing the precision of the prediction intervals. Lastly, the PSR technique contributed the least to the model, reducing MSE, RMSE, and MAE by 11.65%, 6.01% and 10.31%, respectively. The PICP increased by an average of 2.59%, and the PIMW decreased by an average of 138.31%, indicating a limited improvement, but it still optimizes the model's prediction performance to some extent.

**Table 13.** Significance test of ablation experiments for data2.

$p$	MSE	RMSE	MAE	PICP	PIMW
Without NOA	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$	0.6405	0.0008
Without PSR	0.0004	0.0003	$\ll 0.0001$	0.2228	0.0147
Without WT	0.0006	0.0001	0.0005	0.1101	0.0024

As shown in Table 13, the NOA, PSR and WT modules in the proposed model system all passed the significance tests for all error metrics, fully demonstrating the model's significant advantages in improving prediction performance and reducing errors. Although PICP did not show significant differences, PIMW passed the significance test, indicating that the model can effectively reduce the width of prediction intervals. This result is consistent with the conclusions drawn from the data1. Overall, the proposed model system exhibits clear superiority in enhancing prediction accuracy and optimizing interval compactness.

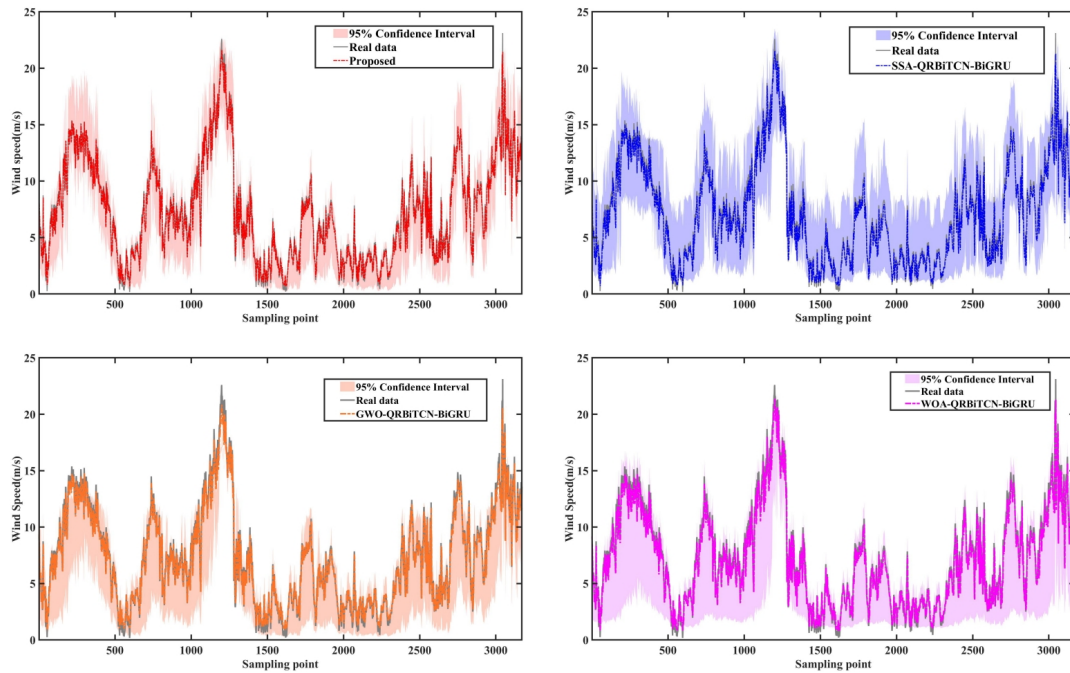
#### 3.4.7. Algorithm comparison based on data2



**Figure 25.** Filtering results for data2.

First, other algorithms were used to optimize the adjustment factor  $\{\alpha, \beta\}$  of the WT model, obtaining the same optimal parameter combination  $\{10, 100\}$  as NOA-WT, with a corresponding fitness value of -27.38092. The optimal adjustment factor  $\alpha = 10$  and  $\beta = 100$  was then incorporated into the WT model to filter the wind speed time series. The filtering results are shown in Figure 25.

Next, the filtered data was subjected to PSR. The Lyapunov exponent was also 0.1862, indicating that the time series is chaotic. Therefore, the QRBiTCN-BiGRU model optimized based on other algorithms was used for interval prediction. The final prediction results are shown in Figure 26.



**Figure 26.** NOA and other algorithms interval prediction for data2.

As shown in Table 14, compared to the model optimized with other algorithms, the model optimized with NOA reduces MSE, RMSE, and MAE by an average of 10.23%, 5.26% and 6.9%, respectively, resulting in smaller prediction errors. The PICP increased by an average of 6.13%, indicating that the model optimized with NOA has better coverage ability, more accurately covering the range of actual values. Additionally, the PIMW decreased by an average of 62.99%, meaning that the NOA not only improves prediction accuracy but also narrows the prediction interval, making the results more concentrated. This enhances the model's credibility and reliability.

**Table 14.** Model evaluation indicators for data2.

Method	MSE	RMSE	MAE	PICP	PIMW
With SSA	0.7133	0.8445	0.6096	0.9640	174.867
With GWO	0.69537	0.83389	0.59963	0.82971	101.9766
With WOA	0.72821	0.85335	0.61661	0.93787	98.2951
With NOA	0.6391	0.7994	0.5665	0.9719	93.08

It can be seen from Table 15 that the model optimized by NOA demonstrates high statistical signifi-

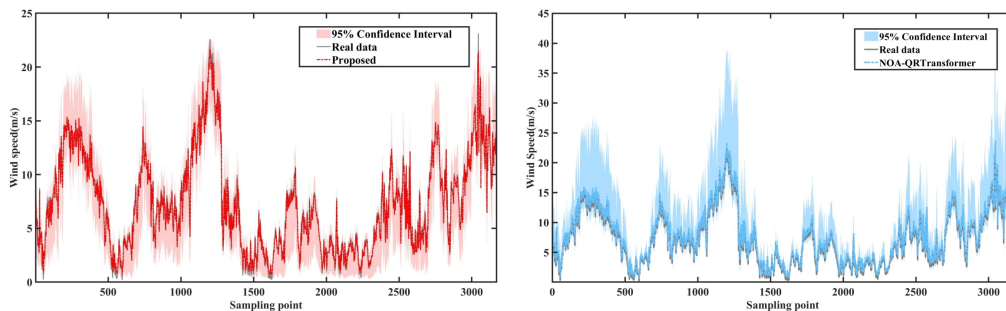
cance in error metrics and PIMW, indicating significant advantages in prediction accuracy and interval width control. However, for interval PICP, the p-values for all algorithms were higher than 0.05, showing no significant differences, which suggests that the performance of the algorithms in interval coverage rate does not differ substantially. This is consistent with the conclusion drawn from data1. Therefore, the model optimized by NOA not only demonstrates significant advantages in error metrics but also shows superior performance in controlling interval prediction width.

**Table 15.** Significance test of algorithm comparison for data2.

$p$	MSE	RMSE	MAE	PICP	PIMW
With SSA	0.0022	0.0016	0.0015	0.2032	0.0212
With GWO	0.0019	0.0015	0.0005	0.6175	0.003
With WOA	0.0093	0.0064	0.0032	0.9484	0.01

#### 3.4.8. Model comparison based on data2

In the experiments of this section, BiTCN-BiGRU was also replaced with Transformer, while the parameter settings and structural design of the remaining modules remained unchanged. The experimental results show that the proposed model not only achieves high-precision prediction but also significantly shortens the inference time, as shown in Figure 27.



**Figure 27.** Model comparison for data2.

From Table 16, it can be seen that MSE, RMSE, and MAE of the proposed model were 0.6391, 0.7994, and 0.5665, respectively, significantly lower than those of NOA-QRTransformer. Also, the proposed model had a PICP of 0.9719 and a PIMW of 93.08, which is much smaller than those of the NOA-QRTransformer. Therefore, the same conclusion as with data1 is drawn: the proposed model outperforms in all metrics, exhibiting higher accuracy and lower uncertainty.

**Table 16.** Model evaluation indicators for data2.

Method	MSE	RMSE	MAE	PICP	PIMW
Proposed	0.6391	0.7994	0.5665	0.9719	93.08
NOA-QRTransformer	0.9250	0.9618	0.7311	0.9189	96.5823

In addition, the indexes of the proposed model passed the significance test, with all p-values lower than 0.05. This indicates that the proposed model not only has high prediction accuracy but also

reduces prediction uncertainty, further verifying the statistical significance of the model performance, as shown in Table 17.

**Table 17.** Significance test of model comparison for data2.

$p$	MSE	RMSE	MAE	PICP	PIMW
Transformer	0.0175	0.0086	0.0048	0.0122	0.001

In 10 experiments, the average inference time of the proposed model was 650 seconds, significantly lower than NOA-QRTransformer's 731 seconds, fully demonstrating its lightweight architecture. By combining shorter inference time with higher prediction accuracy, this model achieves a better balance between performance and resource consumption, demonstrating strong practicality.

#### 4. Conclusions

This article constructs a wind speed interval prediction model and conducts case analysis based on two datasets. The following conclusions are drawn.

First, bidirectional deep learning models fit data better than unidirectional machine learning models. Bidirectional models can more effectively fit the data, significantly reducing prediction errors (MSE, RMSE, MAE) and improving interval coverage (PICP). This indicates that bidirectional models can capture dependencies in time series more comprehensively.

Second, the NOA-WT model reduces deep learning model errors. The NOA-WT model effectively reduces errors in deep learning models by filtering out noise while retaining key data features, thereby improving prediction accuracy. Compared to models without WT, the NOA-WT model significantly reduces prediction errors approximately by 50% to 80%, increases the PICP approximately by 4.21%, and decreases the PIMW approximately by 212%.

Third, the PSR technique improves the accuracy of deep learning models. By reconstructing the embedding dimension of time series, PSR can better capture chaotic characteristics, significantly reducing prediction errors approximately by 25% to 30%. This allows deep learning models to make more accurate predictions, further enhancing model performance.

Fourth, the NOA enhances the robustness of deep learning models. The NOA significantly reduces prediction errors when processing data and maintains high robustness under various data processing conditions. It improves the model's accuracy and prediction interval coverage, especially under multiple model conditions.

Finally, the proposed model demonstrates superior predictive ability. The NOA-QR-BiTCN-BiGRU model proposed in this paper shows lower errors, higher interval coverage, and narrower prediction intervals compared to traditional optimization algorithms and other ensemble models. This proves its superior overall predictive ability and application value in wind speed interval forecasting.

#### Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

This work was supported by the grants from the National Natural Science Foundation (No.12362005), Key Project of Natural Science Foundation of Ningxia (No. 2024AAC02033), Ningxia higher education first-class discipline construction funding project (NXYLXK2017B09), Major Special project of North Minzu University (No. ZDZX201902).

## Conflict of interest

The authors declare there is no conflicts of interest.

## References

1. V. Vigneshwar, S. Y. Krishnan, R. S. Kishna, R. Srinath, B. Ashok, K. Nanthagopal, Comprehensive review of Calophyllum inophyllum as a feasible alternate energy for CI engine applications, *Renewable Sustainable Energy Rev.*, **115** (2019), 109397. <https://doi.org/10.1016/j.rser.2019.109397>
2. M. DeCastro, S. Salvador, M. Gómez-Gesteira, X. Costoya, D. Carvalho, F. J. Sanz-Larruga, et al., Europe, China and the United States: Three different approaches to the development of offshore wind energy, *Renewable Sustainable Energy Rev.*, **109** (2019), 55–70. <https://doi.org/10.1016/j.rser.2019.04.025>
3. J. Chen, G. Q. Zeng, W. Zhou, W. Du, K. D. Lu, Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization, *Energy Convers. Manage.*, **165** (2018), 681–695. <https://doi.org/10.1016/j.enconman.2018.03.098>
4. D. L. Donoho, De-noising by soft-thresholding, *IEEE Trans. Inf. Theory*, **41** (1995), 613–627. <https://doi.org/10.1109/18.382009>
5. D. L. Donoho, I. M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, *J. Am. Stat. Assoc.*, **90** (1995), 1200–1224. <https://doi.org/10.1515/9781400827268.833>
6. S. Wu, L. Jia, Y. Liu, Ultra-short-term wind energy prediction based on wavelet denoising and multivariate LSTM, in *2021 Power System and Green Energy Conference (PSGEC)*, IEEE, (2021), 443–447. <https://doi.org/10.1109/psgec51302.2021.9541909>
7. L. Lian, K. He, Wind power prediction based on wavelet denoising and improved slime mold algorithm optimized support vector machine, *Wind Eng.*, **46** (2022), 866–885. <https://doi.org/10.1177/0309524x211056822>
8. I. Karijadi, S. Y. Chou, A. Dewabharata, Wind power forecasting based on hybrid CEEMDAN-EWT deep learning method, *Renewable Energy*, **218** (2023), 119357. <https://doi.org/10.1016/j.renene.2023.119357>
9. S. E. Kelly, Gibbs phenomenon for wavelets, *Appl. Comput. Harmon. Anal.*, **3** (1996), 72–81. <https://doi.org/10.1006/acha.1996.0006>
10. Y. Lin, J. Cai, A new threshold function for signal denoising based on wavelet transform, in *2010 International Conference on Measuring Technology and Mechatronics Automation*, IEEE, (2010), 200–203. <https://doi.org/10.1109/icmtma.2010.347>

11. L. Su, G. Zhao, R. Zhang, Translation-invariant wavelet de-noising method with improved thresholding, in *IEEE International Symposium on Communications and Information Technology*, IEEE, (2005), 619–622. <https://doi.org/10.1109/iscit.2005.1566931>
12. Z. Peng, S. Peng, L. Fu, B. Lu, J. Tang, K. Wang, et al., A novel deep learning ensemble model with data denoising for short-term wind speed forecasting, *Appl. Comput. Harmon. Anal.*, **207** (2020), 112524. <https://doi.org/10.1016/j.enconman.2020.112524>
13. L. Jing-Yi, L. Hong, Y. Dong, Z. Yan-Sheng, A new wavelet threshold function and denoising application, *Math. Probl. Eng.*, **1** (2016), 3195492. <https://doi.org/10.1155/2016/3195492>
14. Y. Wang, C. Xu, Y. Wang, X. Cheng, A comprehensive diagnosis method of rolling bearing fault based on CEEMDAN-DFA-improved wavelet threshold function and QPSO-MPE-SVM, *Entropy*, **23** (2021), 1142. <https://doi.org/10.3390/e23091142>
15. Y. Qian, Image denoising algorithm based on improved wavelet threshold function and median filter, in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, IEEE, (2018), 1197–1202. <https://doi.org/10.1109/icct.2018.8599921>
16. H. H. Goh, L. Liao, D. Zhang, W. Dai, C. S. Lim, T. A. Kurniawan, et al., Denoising transient power quality disturbances using an improved adaptive wavelet threshold method based on energy optimization, *Energies*, **15** (2022), 3081. <https://doi.org/10.3390/en15093081>
17. Y. Qiao, Q. Li, H. Qian, X. Song, Seismic signal denoising method based on CEEMD and improved wavelet threshold, in *IOP Conference Series: Earth and Environmental Science*, (2021), 012036. <https://doi.org/10.1088/1755-1315/671/1/012036>
18. C. Hu, F. Xing, S. Pan, R. Yuan, Y. Lv, Fault diagnosis of rolling bearings based on variational mode decomposition and genetic algorithm-optimized wavelet threshold denoising, *Machines*, **10** (2022), 649. <https://doi.org/10.3390/machines10080649>
19. F. Ji, X. Cai, J. Zhang, Wind power prediction interval estimation method using wavelet-transform neuro-fuzzy network, *J. Intell. Fuzzy Syst.*, **29** (2015), 2439–2445. <https://doi.org/10.3233/ifs-151944>
20. R. Li, Y. Jin, A wind speed interval prediction system based on multi-objective optimization for machine learning method, *Appl. Energy*, **228** (2018), 2207–2220. <https://doi.org/10.1016/j.apenergy.2018.07.032>
21. Y. Zhang, G. Pan, Y. Zhao, Q. Li, F. Wang, Short-term wind speed interval prediction based on artificial intelligence methods and error probability distribution, *Energy Convers. Manage.*, **224** (2020), 113346. <https://doi.org/10.1016/j.enconman.2020.113346>
22. Z. Gan, C. Li, J. Zhou, G. Tang, Temporal convolutional networks interval prediction model for wind speed forecasting, *Electr. Power Syst. Res.*, **191** (2021), 106865. <https://doi.org/10.1016/j.epsr.2020.106865>
23. Y. Liu, H. Qin, Z. Zhang, S. Pei, Z. Jiang, Z. Feng, et al., Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model, *Appl. Energy*, **260** (2020), 114259. <https://doi.org/10.1016/j.apenergy.2019.114259>



24. X. Yuan, C. Chen, M. Jiang, Y. Yuan, Prediction interval of wind power using parameter optimized Beta distribution based LSTM model, *Appl. Soft Comput.*, **82** (2019), 105550. <https://doi.org/10.1016/j.asoc.2019.105550>
25. N. Pei, Y. Wu, R. Su, X. Li, Z. Wu, R. Li, et al., Interval prediction of the permeability of granite bodies in a high-level radioactive waste disposal site using LSTM-RNNs and probability distribution, *Front. Earth Sci.*, **10** (2022), 835308. <https://doi.org/10.3389/feart.2022.835308>
26. K. Zhang, X. Yu, S. Liu, X. Dong, D. Li, H. Zang, et al., Wind power interval prediction based on hybrid semi-cloud model and nonparametric kernel density estimation, *Energy Rep.*, **8** (2022), 1068–1078. <https://doi.org/10.1016/j.egy.2022.02.094>
27. J. Wang, S. Wang, B. Zeng, H. Lu, A novel ensemble probabilistic forecasting system for uncertainty in wind speed, *Appl. Energy*, **313** (2022), 118796. <https://doi.org/10.1016/j.apenergy.2022.118796>
28. X. Peng, H. Wang, J. Lang, W. Li, Q. Xu, Z. Zhang, et al., EALSTM-QR: Interval wind-power prediction model based on numerical weather prediction and deep learning, *Energy*, **220** (2021), 119692. <https://doi.org/10.1016/j.energy.2020.119692>
29. J. Wang, S. Wang, Z. Li, Wind speed deterministic forecasting and probabilistic interval forecasting approach based on deep learning, modified tunicate swarm algorithm, and quantile regression, *Renewable Energy*, **179** (2021), 1246–1261. <https://doi.org/10.1016/j.renene.2021.07.113>
30. M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, *Knowl.-Based Syst.*, **262** (2023), 110248. <https://doi.org/10.1016/j.knosys.2022.110248>
31. J. Zhou, X. Lu, Y. Xiao, J. Su, J. Lyu, Y. Ma, et al., Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022, preprint, arXiv:2208.04360.
32. L. Tang, J. Liang, CC method to phase space reconstruction based on multivariate time series, in *2011 2nd International Conference on Intelligent Control and Information Processing*, IEEE, (2011), 438–441. <https://doi.org/10.1109/icicip.2011.6008282>
33. A. Wolf, J. B. Swift, H. L. Swinney, J. A. Vastano, Determining Lyapunov exponents from a time series, *Physica D*, **16** (1985), 285–317. <https://doi.org/10.1007/bfb0086675>
34. J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. <https://doi.org/10.1080/21642583.2019.1708830>
35. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. <https://doi.org/10.1201/9781003206477-8>
36. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)