EAI Endorsed Transactions
on Energy Web

Research Article **EAI.EU**

Long-term Wind Power Forecasting Based on Bayesian Optimization-Based CNN-GRU Model

Lin Wang[1], Ying Yang[2,*]

[1]School of Municipal and Geomatics Engineering, Hunan City University, Yiyang ,413000, China
[2]Department of Discipline Inspection and Supervision, Hunan Vocational College of Engineering, Changsha, 410151, China

Abstract

The safe and stable operation of the power grid, the optimization of power system dispatching, and the improvement of the economic benefits of wind farms all rely on long-term wind power prediction. Long-term wind power prediction requires handling complex time series and spatial dependencies. Insufficient data availability, accumulation of prediction errors, complexity of the model, limitations of computing resources, volatility of wind energy, and the accuracy of long-term numerical weather forecasts can all affect the accuracy of predictions. Therefore, long-term wind power prediction is more difficult than short-term prediction. This paper uses the Bayesian optimization-based CNN-GRU combined model (BO-CNN-GRU) to predict the long-term wind power in a low-wind-speed wind farm. CNN is good at extracting local spatial features from the input data. GRU focuses on modeling long-term dependencies in time series. The two complement each other. The instability and nonlinearity of wind power can be handled effectively. Bayesian optimization can handle datasets with sharp fluctuations and avoid overfitting. Compared with the CNN-GRU model, the BO-CNN-GRU model has higher prediction accuracy: the mean square error (MSE) is reduced by an average of 19.27%, the root mean square error (RMSE) is reduced by an average of 11.10%, and the mean absolute error (MAE) is reduced by an average of 17.24%. The generalization ability of the model has also been enhanced: the MSE, RMSE and MAE values have decreased by 20.45%, 10.45%, and 17.85%, respectively. Meanwhile, the model has good robustness and strong seasonal universality. The new model has a good application effect in our case.

Keywords: Long-term wind power forecasting; Bayesian Optimization; CNN-GRU hybrid model; Low-wind-speed wind farm

Received on 07 January 2026, accepted on 11 March 2026, published on 16 September 2026

doi: 10.4108/ew.14197

1. Introduction

In recent years, wind power has been an important clean energy source. However, due to the intermittent and fluctuating characteristics of its power generation capacity, the stable operation of the power grid is facing severe tests [1,2]. If the long-term wind power can be accurately predicted, the stable operation of the power grid and the dispatching of the power system can both be guaranteed, the operating costs of wind farms will decrease, and economic benefits will increase [2,3].

At present, approximately 87.22% of the research results on wind power prediction are related to ultra-short-term and short-term wind power prediction [4], while there are few studies on long-term wind power prediction. Long-term wind power prediction requires handling more complex time series and spatial dependencies. The key challenges include: insufficient data quality and availability [5], forecast accuracy and error accumulation issues [6], increased model complexity and computational resource demands [7], the uncertainty of wind power [8], the accuracy of long-term numerical weather prediction [9], and model adaptability and generalization capability. So long-term wind power forecasting is more difficult than

*Corresponding author. Email: 493662400@qq.com

short-term forecasting, and corresponding prediction research is relatively insufficient.

There are four types of long-term wind power prediction methods. The first type is statistical methods, including ARIMA and its variants. The accuracy of the methods heavily depends on historical data, which makes their long-term forecasting results not good. Thus, they often need to be combined with other models [10]. The second type is machine learning, including SVM RF, and ANN, etc [11-13]. The third type is deep learning, represented by CNN, RNN, and LSTM. Among these, LSTM is relatively widely applied in long-term forecasting [14,15]. The last type is hybrid models. Researchers often combine the aforementioned methods to improve robustness. Examples include integrating wavelet transform with LSTM to handle seasonal and regional effects [6], Principal component analysis is used for dimensionality reduction before establishing a low-rank regularization model [16], and multi-model fusion schemes [17]. Although these hybrid approaches require larger data volumes and higher computational power, they can merge the strong points of different models. So these hybrid approaches demonstrate better accuracy and stability under extreme weather conditions. Long-term wind power prediction requires data over a longer time. The prediction needs to consider the influence of various meteorological factors and long-term trends. The application of combined models can improve prediction accuracy.

This paper intends to use BO-CNN-GRU for long-term wind power prediction research. CNN is good at extracting local spatial features from input data such as wind speed, air pressure, temperature, etc. GRU, on the other hand, focuses on modeling long-term dependencies in time series. The combination of the two can effectively handle the instability and nonlinearity of wind power. Bayesian optimization establishes a probability model in the hyperparameter space, which is more robust for datasets with sharp fluctuations and can avoid overfitting.

The BO-CNN-GRU model was applied to predict the long-term wind power of a low-wind-speed wind farm in the Jianghan Plain. First, normalize the data, then extract features through CNN and input them into GRU to capture the temporal relationship. Secondly, Bayesian optimization is utilized to adjust hyperparameters and optimize model performance. Finally, the trained model is used for prediction, and the results are reverse-normalized to achieve accurate prediction. The results of the case analysis show that this method has smaller errors, higher prediction accuracy, and stronger generalization ability than other wind power prediction methods.

2. Methodology

2.1. KAN

The mathematical basis of KAN networks is the Kolmogorov-Arnold representation theorem [18]: Any continuous multivariable function can be expressed as a finite combination and superposition of univariable functions. For any continuous function $f:[0,1]^n \to \mathrm{R}$, there exist continuous univariate functions $\Phi_{q,p}$ and Ψ_q:

$$f(x_1, x_2 ..., x_n) = \sum_{q=1}^{2n+1} \Psi_q \left(\sum_{p=1}^{n} \Phi_{q,p}(x_p)\right) \tag{1}$$

KAN abandons the fixed activation mechanism of traditional neural networks and instead utilizes a series of single-variable function networks with learnable characteristics to fit complex multivariate target functions, thereby achieving the precise reconstruction of any continuous function. Compared to classical neural networks, this new architecture just does a better job at generalizing and fitting the data when dealing with high-dimensional nonlinear mapping tasks.

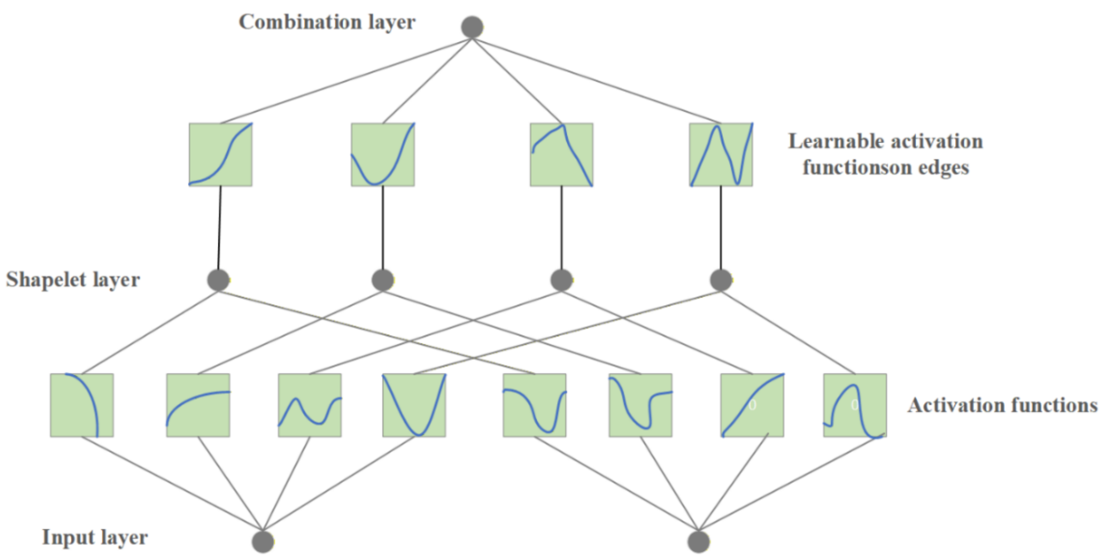

Figure 1. KAN model structure

KAN uses B-spline functions as the fundamental building blocks to represent univariate functions $\Phi_{q,p}$ and Ψ_q. B-splines are piecewise polynomial functions characterized by their local support and smoothness, making them particularly suitable for function approximation tasks. In KAN, each univariate function is represented in the following form: where K is the number of B-spline basis functions used.

$$\Phi(x) = \sum_{i=1}^{K} w_i B_i(x) \tag{2}$$

Where $B_i(x)$ is the number of B-spline basis functions used. w_i is the learnable weight parameter, and K is the number of B-spline basis functions used. The schematic diagram of a two-layer KAN is shown in Figure 1.

2.2. CNN

CNN [19] is a neural network architecture initially applied mainly in image recognition and processing. CNN draws on the working principle of the human visual system and extracts local feature information from data through continuous convolution operations. CNN structure includes components such as an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer (Figure 2). In CNN, each convolution kernel corresponds to a specific feature extraction pattern. When processing time series data, we usually adopt a one-dimensional convolutional neural network (1DCNN), which extracts local features and reduces the dimensionality of raw data by extracting local features of the input data with convolution operations. The main function of the pooling layer is to reduce the dimensionality of the features extracted by the convolutional layer, which not only simplifies the network structure but also retains the most critical feature information. At the end of the network, the fully connected layer is responsible for weighting the features and outputting results, and these layers usually use ReLU as the activation function. The basic calculation process of CNN is like this:

$$y_{conv} = f(X * w_p + b_p) \tag{3}$$

Where y_{conv} denotes the output of the convolutional layer, f is the activation function of the convolutional layer, X is the matrix composed of input data, w_p is the weight matrix of the convolutional layer, and b_p is the bias vector of the convolutional layer.

$$y_{pool} = \delta(y_{conv}) \tag{4}$$

y_{pool} represents the output of the pooling layer, and δ denotes the pooling method used in the pooling layer.

$$y_{fc} = g(W_s y_{pool} + b_s) \tag{5}$$

y_{fc} denotes the output of the fully connected layer, W_s is the weight matrix of the fully connected layer, b_s is the bias vector of the fully connected layer, and g is the activation function of the fully connected layer.

2.3. GRU

GRU represents a significant advancement in RNN architecture [20]. As a variant of LSTM networks, GRU can capture dependencies in different time scales while solving the inherent vanishing gradient problem in traditional RNN models.

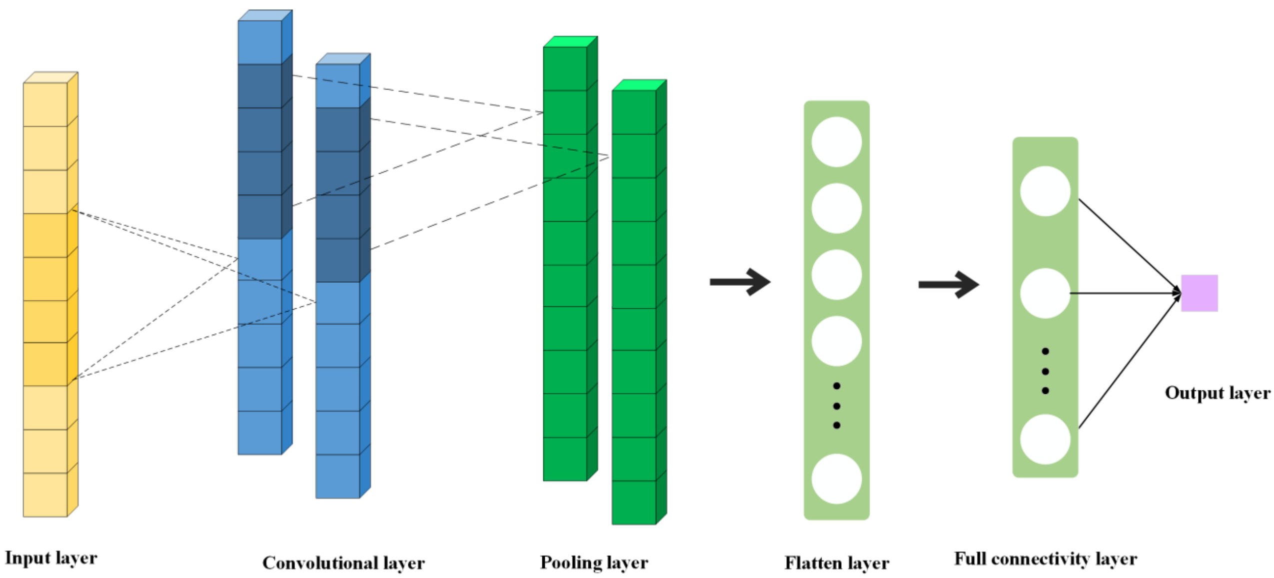

Figure 2. One-dimensional convolutional neural network structure

The GRU architecture employs two main gating mechanisms: the reset gate r_t and the update gate z_t, as illustrated in Figure 3. For an input sequence x, the GRU computes the hidden state h_t at time step t as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{6}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{7}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \tag{8}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{9}$$

Where σ denotes the sigmoid activation function, tanh is the hyperbolic tangent function, and ⊙ represents element-wise multiplication (Hadamard product). The terms W_r, W_z, W_h, U_r, U_z, and U_h are weight matrices, while b_r, b_z, and b_h are bias vectors.

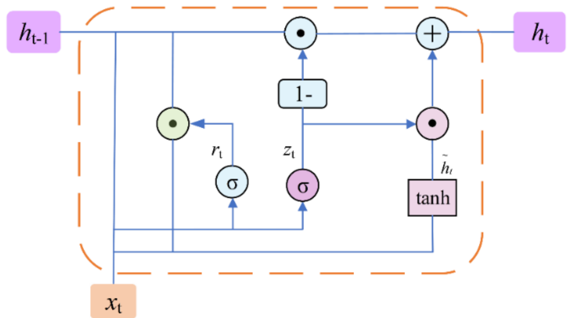

Figure 3. GRU structure

2.4. CNN-GRU

CNN excel in regression prediction by enabling local feature extraction and parameter sharing, effectively capturing spatial correlations in data, which is suited for processing high-dimensional datasets. Its drawbacks include the difficulty in modeling long-term temporal dependencies, sensitivity to input data shifts, and potential oversight of global temporal information. The GRU offers higher computational efficiency than LSTM. However, its ability to extract local features is relatively weak. In processing high-dimensional data, the large number of parameters may increase the overfitting risk.

For large-scale data, CNN-GRU [21] combines the strengths of both architectures. It combines the local feature extraction capability of CNN and the long-range time series modeling capability of GRU. The CNN-GRU architecture is shown in Figure 4. This hybrid architecture achieves feature refinement and long-term dynamic modeling at the same time, so it is suitable for high-dimensional, large-sample regression tasks. Additionally, it helps reduce overfitting risks through staged feature learning.

3. Hyperparameter Optimization Based on the Bayesian Optimization Algorithm

3.1. Bayesian Optimization Algorithm

Bayesian optimization employs an agent model (typically a Gaussian process) [22,23] to model the objective function. The Gaussian process is defined by the mean function m(x) and the covariance function $K(x, x^{'})$.The mean function provides the mean estimate of the objective function at each point.

First, N_{init} configuration points are randomly sampled within a predefined hyperparameter space, and the model is trained at these points to obtain the initial observation dataset $D = \{(x_i, y_i)\}_{i=1}^{N_{init}}$, where x_i denotes the hyperparameter configuration and y_i is the corresponding performance metric (e.g., validation error). Subsequently, Gaussian Process Regression (GPR) is employed to fit D, yielding the posterior distribution of the objective function f(x): the mean function μ(x) and the covariance function σ2(x). We then build an acquisition function α(x) from the posterior distribution, which can strike a balance between exploration and exploitation.

$$x_{next} = \arg\max \alpha(x) \tag{10}$$

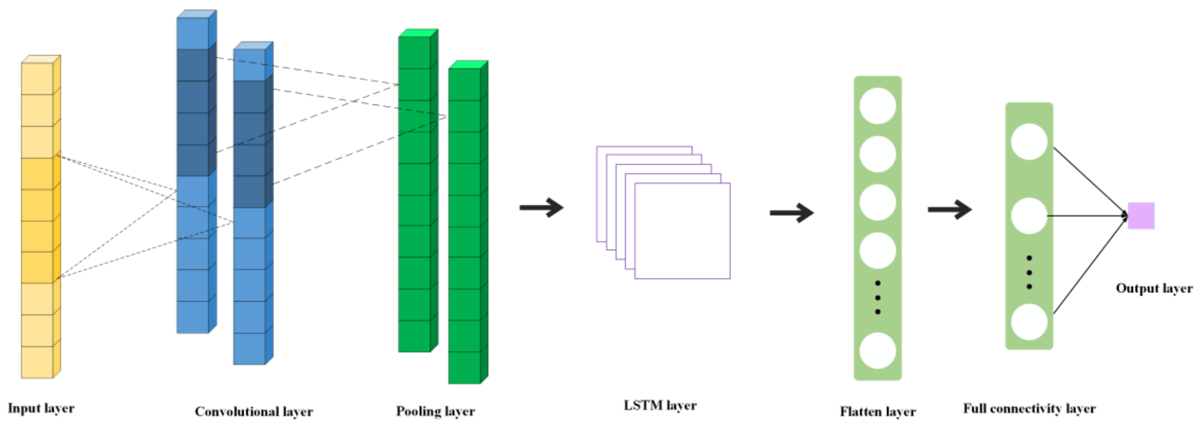

Figure 4. CNN-GRU structure

The hyperparameter configuration is obtained for the next evaluation. The model is then trained at x_{next} to measure its performance $y_{nest} = f(x_{next})$, and the new observation (x_{next}, y_{next}) is incorporated into the dataset D. The above process of Gaussian Process (GP) fitting and acquisition function optimization is repeated iteratively until a predefined stopping criterion is met (e.g., maximum number of iterations, time budget, or performance convergence threshold). Finally, the configuration with the best historical performance $x^* = \arg\min_{(x,y)\in D}$. We select y as the optimal hyperparameter set and use it to conduct a full training and validation run. The optimization process is shown in Figure 5.

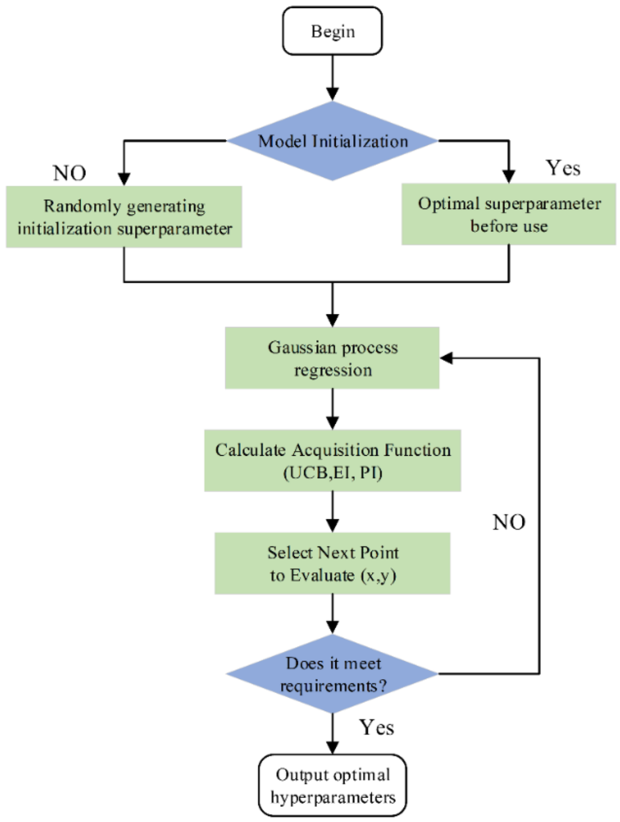

Figure 5. Bayesian optimization process

3.2. Hyperparameter Optimization

During the building of deep learning models, the hyperparameter selection greatly affects model performance. Table 1 lists their optimization ranges. The kernel size determines the receptive field of filters in convolutional neural networks (CNNs), directly affecting the ability to get features. An excessively small parameter size fails to capture multidimensional feature interaction patterns, and an excessively large parameter size leads to exponential growth in parameter quantity. An appropriate learning rate ensures rapid model convergence and prevents trapping at local minima. Batch size affects the training stability and the training speed, while also directly impacting memory requirements. As a regularization technique, Dropout discards neurons to prevent overfitting, and its dropout rate must be finely tuned to maintain generalization ability [24].

Table 1. Ranges of hyperparameter values

Hyperparameter	Tunning range
Kernel size	{3,6}
Learning rate	{1e-4,1e-3}
Batch size	{32,64}
Dropout	{0.05, 0.5}
CNN layer units	{32, 128}
GRU layer units	{32, 128}
GRU layers	{1, 3}

For RNN, particularly GRU, the number of units and layers in GRU layers determines the ability to get temporal dependencies. Through Bayesian optimization, we can explore the hyperparameter space with reasonable efficiency. Depending on the setup, this may lead to better model performance and, in some cases, reduce unnecessary computational cost [19].

4. Database Establishment

4.1. Data Preprocessing

For the dataset collected from 19 different base stations, the data preprocessing process primarily includes the following steps: First, raw data is cleaned by removing irrelevant non-feature columns; subsequently, the data from different base stations are divided into training date, validation date, and test date, and the ratio is 7:1.5:1.5. To mitigate the impact of feature dimensionality differences on model performance, we employ StandardScaler for feature and target variable normalization. The standardized parameters are fitted using training set data, and implement the same transformations on validation and test sets to prevent the data leakage. Given the characteristics of time-series data, we implemented a sliding-window structure that uses a continuous sequence of 20 time steps as input to predict the target value at the next time point.

4.2. Feature Selection

Figure 6 presents a Pearson correlation heat map illustrating the relationship between active power and environmental parameters of wind turbines. The prominent red areas in the figure indicate a strong positive correlation, whereas the areas near white or light blue represent weak or negative correlations.

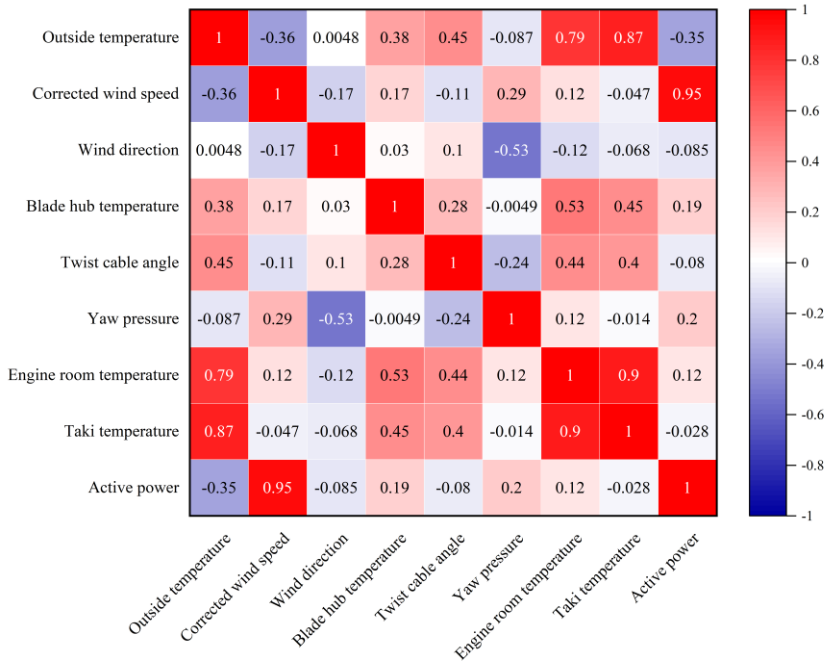

Figure 6. Heatmap of Input Features

As shown in the figure, the most significant positive correlation is observed between adjusted wind speed and active power (with a correlation coefficient as high as 0.95), confirming wind speed's pivotal role as the primary driver of wind power generation. Furthermore, the external temperature positively correlated with the cabin temperature and the tower base temperature, indicating that the ambient temperature directly and significantly influences the internal temperature of the unit.

In addition to the core factors driving power generation, the matrix also reveals correlations among other parameters. For instance, active power exhibits a moderate negative correlation with ambient temperature, which may be attributed to air density or unit performance characteristics. Meanwhile, yaw pressure exhibits a moderate negative correlation with wind direction, reflecting the mechanical response of the turbine unit during yaw adjustments to capture optimal wind energy directions. The blade pitch temperature and pitch angle also exhibit a certain positive correlation with ambient and nacelle internal temperatures, demonstrating that the operational status of these critical components is jointly influenced by both external and internal thermal environments. Overall, this matrix provides a comprehensive perspective that facilitates understanding how various subsystems of wind turbines interact with external environmental factors and ultimately influence their power generation performance.

4.3. Evaluation Metrics

We select four performance evaluation metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2, R-Square). Among them, the smaller the values of the first three error indicators are, the higher the accuracy of the prediction model is. While the closer the coefficient of determination R^2 is to 1, the better the fitting effect of the model is. The formulas for these evaluation indicators are like those:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{11}$$

$$RMSE = \sqrt{\frac{1}{m}(y_i - \hat{y}_i)^2} \tag{12}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}| y_i - \hat{y}_i | \tag{13}$$

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2} \tag{14}$$

Here, $\hat{y}_i$ denotes the predicted value, y_i represents the true value, $\bar{y}_i$ indicates the mean of true values, and m stands for the number of samples.

5. Results and Discussion

5.1. Prediction Results

We collected wind power data from 19 base stations (FJ01–FJ19) and selected FJ16–FJ19 as the test data to evaluate the models. Tables 2 to 5 present the comparison of evaluation metrics for the four models.

Table 2. Comparison of evaluation indexes of each model on FJ1

Models	MSE	RMSE	MAE	R^2
LSTM	46617.78	215.91	148.43	0.97
GRU	47337.01	217.57	158.35	0.97
LSTM-KAN	600522.81	774.93	679.22	0.64
GRU-KAN	572885.33	756.89	661.97	0.65
CNN-GRU	45307.21	212.85	148.69	0.97
BO-CNN-GRU	38127.31	195.26	123.44	0.98

Table 3. Comparison of evaluation indexes of each model on FJ6

Models	MSE	RMSE	MAE	R^2
LSTM	63724.58	252.44	162.78	0.96
GRU	64497.33	253.96	165.28	0.96
LSTM-KAN	640709.19	800.44	681.53	0.62
GRU-KAN	630597.78	794.10	679.99	0.63
CNN-GRU	65333.14	255.60	164.25	0.96
BO-CNN-GRU	57180.04	239.12	140.56	0.97

Table 4. Comparison of evaluation indexes of each model on FJ11

Models	MSE	RMSE	MAE	R^2
LSTM	72759.94	269.74	176.18	0.94
GRU	60847.54	246.67	167.56	0.95
LSTM-KAN	551453.51	742.59	659.75	0.57
GRU-KAN	564786.15	751.52	669.68	0.56
CNN-GRU	45307.21	212.85	148.69	0.97
BO-CNN-GRU	59397.66	243.72	145.69	0.95

Table 5. Comparison of evaluation indexes of each model on FJ17

Models	MSE	RMSE	MAE	R^2
LSTM	55482.96	235.55	157.92	0.97
GRU	58078.45	240.99	166.88	0.96
LSTM-KAN	604017.42	777.18	659.06	0.62
GRU-KAN	587191.02	766.28	654.16	0.63
CNN-GRU	62103.85	249.21	165.38	0.96
BO-CNN-GRU	47962.74	219.00	139.73	0.97

Figure 7 illustrates the loss curve of the BO-CNN-GRU model during training. The figure demonstrates that during the initial training phase (the first 5 epochs), both training loss and validation loss fall sharply. It shows that the model have the ability to capture dominant features quickly in time-series data and achieve swift convergence. As training progressed, both the curves followed each other at a steady pace of decline with no loss recovery or convergence stagnating in the validation set. It shows that the model can prevent the typical overfitting and underfitting challenges of deep learning. After 20 rounds of training, the loss value gradually stabilized (around 0.03), resulting in an extremely small gap between the training and validation loss. Overall, the convergence curve stability confirms that the combination of hyperparameters found by the Bayesian optimization is effective. The existing model not only becomes a good example of the high-fitting accuracy on training data but also has an outstanding fitting accuracy on new data.

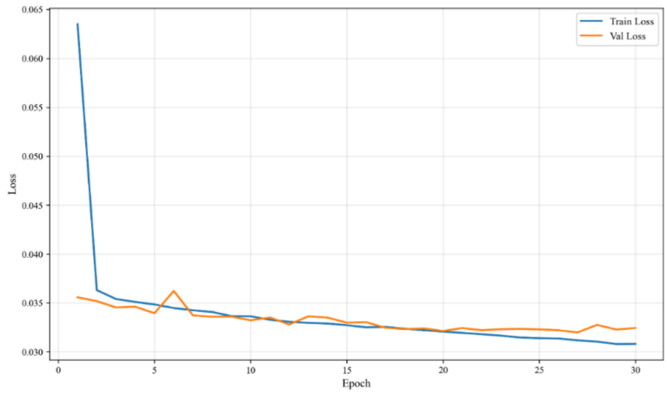

Figure 7. Training and validation loss curves of the CNN-GRU model after Bayesian optimization

In our study, the predictive performance of the BO-CNN-GRU model is shown in Figures 8 to 11. The left figure compares the trends of actual and predicted values, with the blue solid line representing the observed values and the red dashed line representing the model's predictions. With this chart, we will be able to find out how well predictions are performed by the model at various sample points and whether they can show the whole trend. The right graph is a scatter plot of prediction accuracy. Each point in the graph represents a sample, with the color intensity indicating the absolute value of the prediction error—the lighter the color, the larger the error. The figure also displays RMSE and R^2, indicating high prediction accuracy and excellent fit.

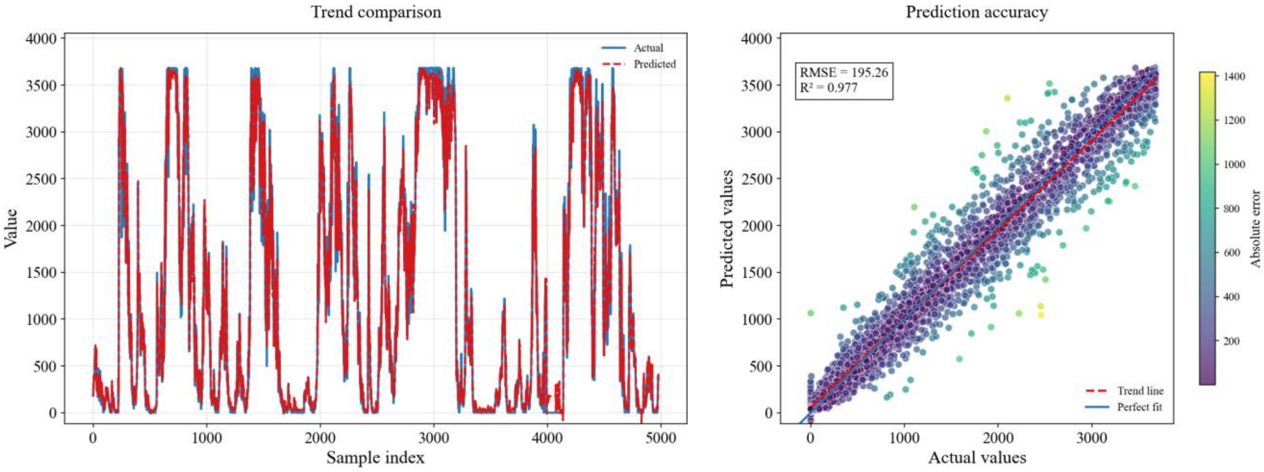

Figure 8. Prediction results of BO-CNN-GRU on FJ16

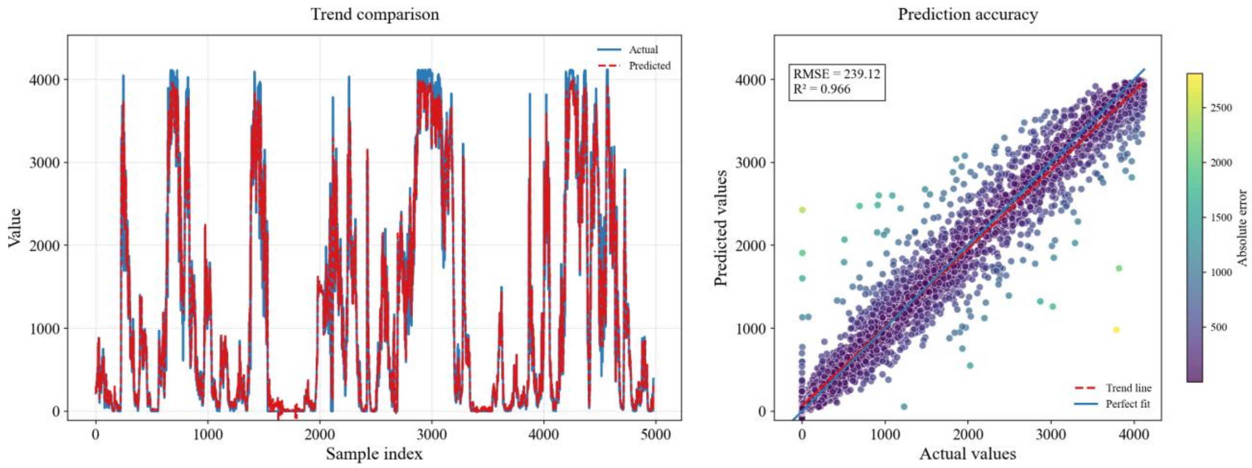

Figure 9. Prediction results of BO-CNN-GRU on FJ17

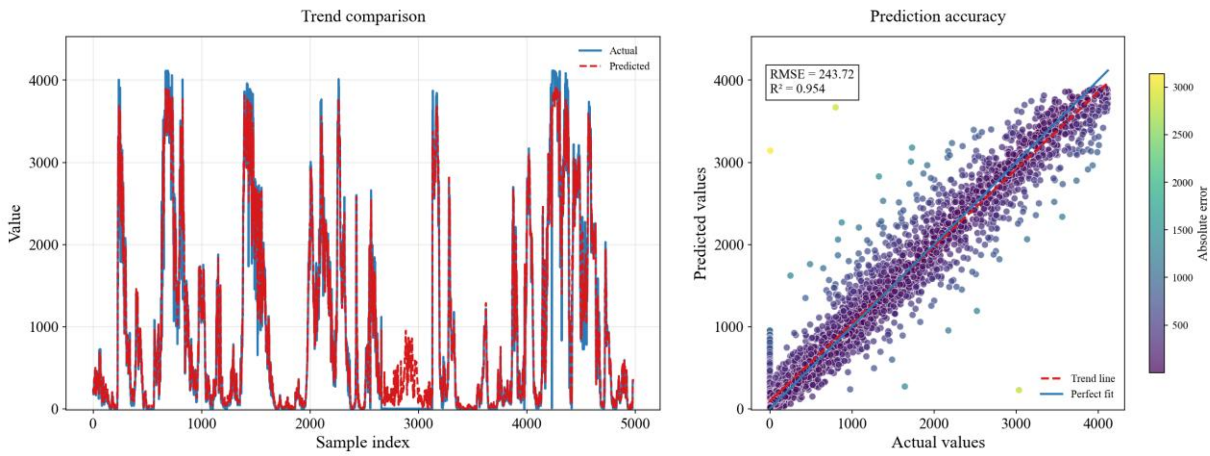

Figure 10. Prediction results of BO-CNN-GRU on FJ18

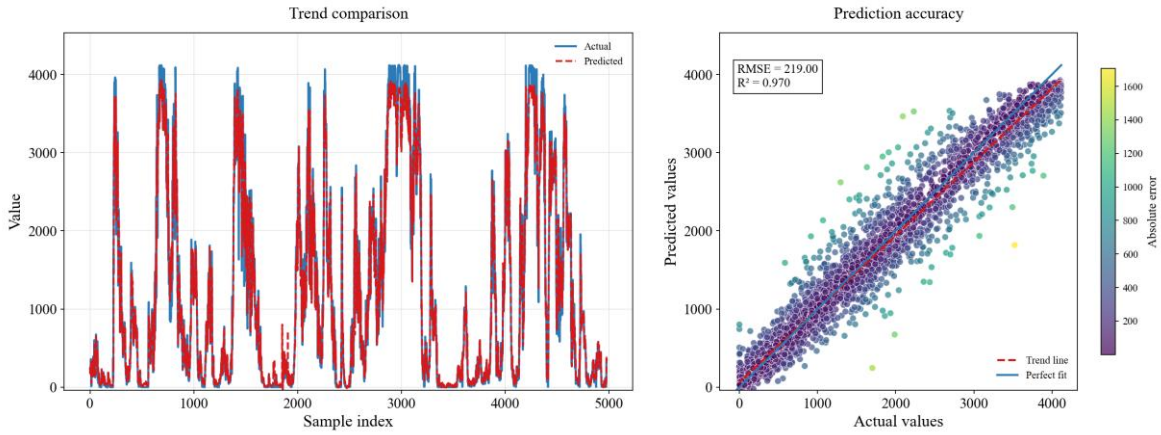

Figure 11. Prediction results of BO-CNN-GRU on FJ19

5.2. Predictive Performance Evaluation

This study aggregated all data from FJ16 to FJ19 and assessed the models' performance on this comprehensive dataset. In Table 6, we can see that the BO-CNN-GRU model demonstrates strong advantages across all key quantitative evaluation metrics. Compared with other models (LSTM, GRU, CNN-GRU), BO-CNN-GRU has the lowest values for MSE, RMSE, and MAE. This directly reflects its minimal average deviation between predicted and actual values, resulting in the highest prediction accuracy. And the BO-CNN-GRU model has the highest coefficient of determination (R^2) of 0.968. It demonstrates that the new model has a superior ability to explain the variability in the original data. The new model has a good fit to the data.

Table 6. Comparison of evaluation indexes of all data of each model in FJ16~FJ19

Models	MSE	RMSE	MAE	R^2
LSTM	55482.96	235.55	157.92	0.97
GRU	58078.45	240.99	166.88	0.96
LSTM-KAN	604017.42	777.18	659.06	0.62
GRU-KAN	587191.02	766.28	654.16	0.63
CNN-GRU	62103.85	249.21	165.38	0.96
BO-CNN-GRU	47962.74	219.00	139.73	0.97

Figure 12 presented in the form of a violin plot, visually illustrates the differences between the distribution of the combined prediction results from various models and the original data. The prediction distribution of the BO-CNN-GRU model (blue violin plot) exhibits the highest similarity to the original data distribution (green violin plot) in terms of overall shape, data density distribution, and prediction range. Although the average prediction value of the GRU model (1122.38) is closest to the original data mean (1120.91), the low error metrics in Table 6 demonstrate that the BO-CNN-GRU model not only delivers predictions close to the true values but, more importantly, successfully captures the overall distribution characteristics and patterns of variation in the original data—a critical capability for understanding and predicting data variability in practical applications. Comprehensive quantitative metrics and distribution comparative analyses conclusively demonstrate that the BO-CNN-GRU model exhibits remarkable superiority and enhanced reliability in complex time-series prediction tasks, owing to its Bayesian-optimized parameter tuning and the synergistic advantages of its CNN and GRU architectures.

5.3. Hyperparameter Analysis

In Figure 13, the sign and absolute value of the correlation coefficients in the figure intuitively reflect the trend of the impact of parameter adjustment on model error. Since the optimization goal is to minimize the validation loss, parameters showing a negative correlation (green to blue regions) indicate that an increase in their values helps reduce the error, while a positive correlation (yellow to red regions) indicates that an increase in their values leads to an error increase.

Analysis of the heat map reveals: error

(1) Strong negative correlations: Learning rate (learning_rate, r = −0.690), CNN output channel count (cnn_out_channel, r = −0.665), and GRU hidden layer dimension (gru_hidden_dim, r = −0.546) all exhibit strong negative correlations.

(2) Positive correlation: batch size (batch_size, r=0.400), number of GRU layers (gru_layers, r=0.286), and dropout rate (dropout_rate, r=0.243) are positively correlated with error.

(3) Parameters with low sensitivity: The correlation coefficient of kernel size (kernel_size, r=0.032) is close to 0, indicating that changing the receptive field size of the

1D convolution has no significant linear impact on the final prediction error.

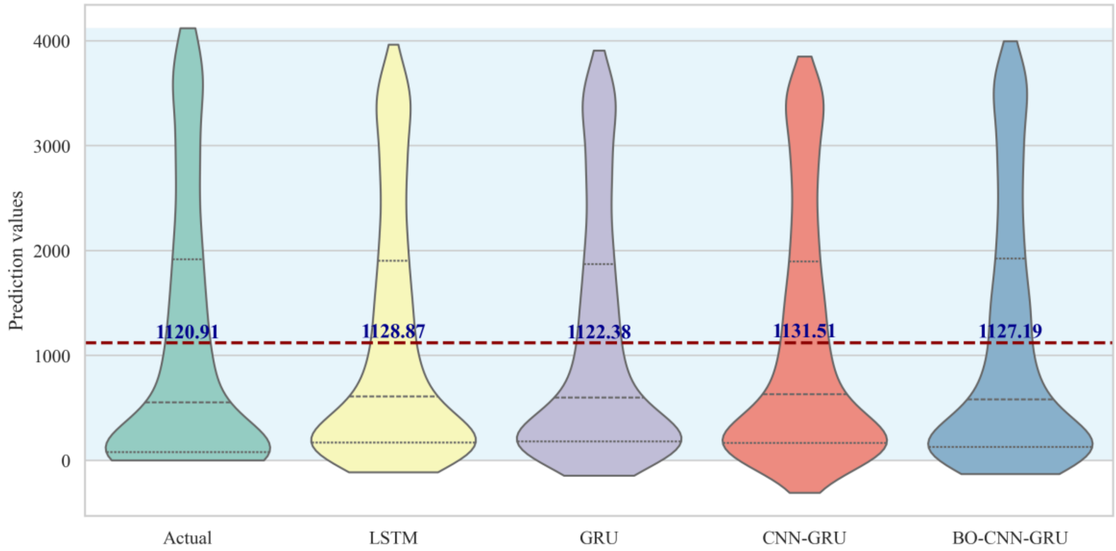

Figure 12. Violin diagram comparing the sum of prediction results of each model with the original data

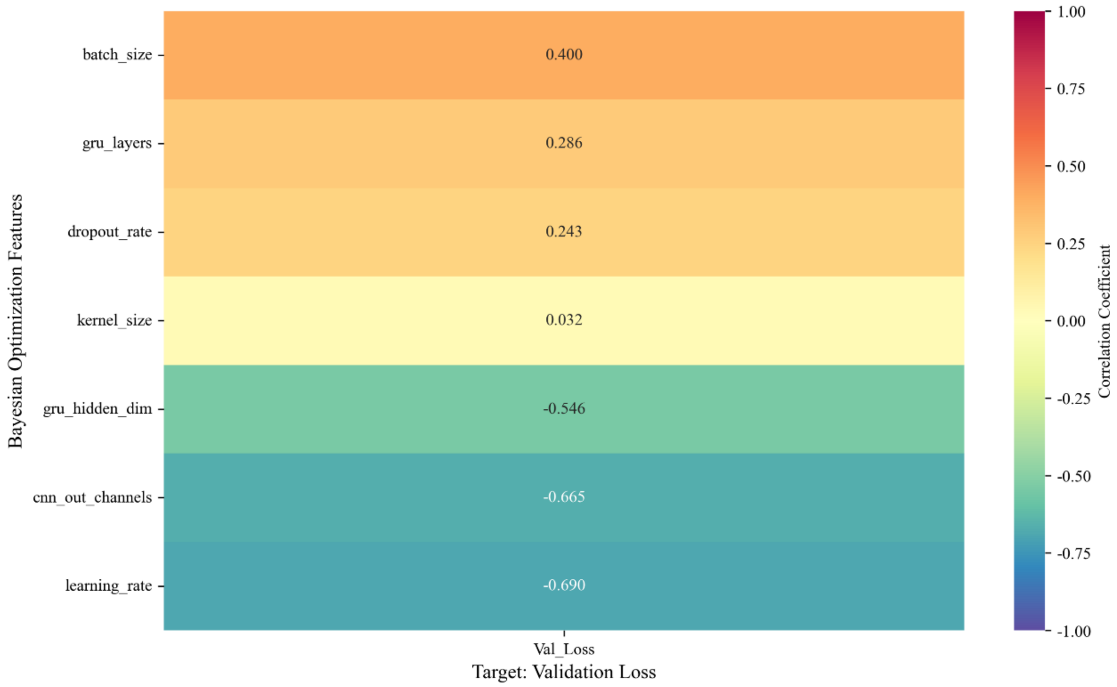

Figure 13. Pearson correlation heatmap between CNN-GRU hyperparameters and validation loss during Bayesian optimization

Therefore, the CNN-GRU model prefers a “Wide and Shallow” network architecture in structural design (high number of channels, high hidden dimension, and fewer layers), and relies on a smaller batch size in the training strategy.

6. Conclusion

Aiming to wind power prediction of a low-wind-speed wind farm in the Jianghan Plain, we propose the BO-CNN-GRU model. Experimental verification shows that the model exhibits significant advantages in prediction accuracy, generalization ability, and robustness.

The main conclusions of this paper are as follows:

(1) Improvement in Prediction Accuracy: The BO-CNN-GRU model has achieved a significant improvement in prediction accuracy. Specifically, the MSE of the BO-CNN-GRU model on the FJ16, FJ17, FJ18, and FJ19 test sets is reduced by 15.95%, 12.83%, 25.53%, and 22.76% respectively compared to the CNN-GRU model. The RMSE is reduced by 8.22%, 6.65%, 16.47%, and 13.07% respectively, and the MAE is reduced by 17.85%, 14.44%, 20.81%, and 15.85% respectively.

(2) Enhanced Generalization Ability: In the comprehensive evaluation of all data from FJ16 to FJ19, the MSE, RMSE, and MAE of the BO-CNN-GRU model reach the lowest values of 50666.93, 225.09, and 137.35 respectively, which are 20.45%, 10.45%, and 17.85% lower than those of the unoptimized CNN-GRU model.

(3) Good Robustness: The BO-CNN-GRU model shows better robustness when facing the uncertainty and volatility of wind power. In extreme weather conditions, this model can more accurately capture the change trend of wind power, thus providing a more reliable prediction for the operation and management of wind farms.

Acknowledgements

Not applicable.

References

[1] Yu, Y., Wang, J., Xu, D., Wang, Z., Yang, T. Long-Term Wind Power Forecasting Using Series Decomposition and Multi-Scale Attention Mechanism. 2024 5th International Conference on Power Engineering (ICPE); 2024. p. 663–668. https://doi.org/10.1109/icpe64565.2024.10929426

[2] Ramadevi, B., Kasi, V. R., Bingi, K. Hybrid LSTM-Based Fractional-Order Neural Network for Jeju Island's Wind Farm Power Forecasting. Fractal and Fractional. 2024;8(3):149. https://doi.org/10.3390/fractalfract8030149

[3] Liu, Z., Guo, H., Zhang, Y., Zuo, Z. A Comprehensive Review of Wind Power Prediction Based on Machine Learning: Models, Applications, and Challenges. Energies. 2025;18(2):350. https://doi.org/10.3390/en18020350

[4] Demolli, H., Dokuz, A. S., Ecemis, A., Gokcek, M. ''Wind power forecasting based on daily wind speed data using machine learning algorithms,''. Energy Convers. Manage. 2019;198:111823.

[5] Xiao, S., Lu, J., Li, D., Yu, H., Li, M., Liu, J. Long-term wind power forecasting model by multi meteorological variables based on data compensation. 2017 2nd International Conference on Power and Renewable Energy (ICPRE); 2017. p. 451–455. https://doi.org/10.1109/icpre.2017.8390576

[6] Chen, K.-S., Lin, T.-Y., Lin, K.-P., Chang, P.-T., Wang, Y.-C. Developing a Novel Long Short-Term Memory Networks with Seasonal Wavelet Transform for Long-Term Wind Power Output Forecasting. International Journal of Computational Intelligence Systems. 2023;16(1):191. https://doi.org/10.1007/s44196-023-00371-x

[7] Dhaka, P., Sreejeth, M., Tripathi, M. M. A Survey of Artificial Intelligence Applications in Wind Energy Forecasting. Archives of Computational Methods in Engineering. 2024;31(8):4853–4878. https://doi.org/10.1007/s11831-024-10182-8

[8] Borunda, M., Ramírez, A., Garduno, R., García-Beltrán, C., Mijarez, R. Enhancing Long-Term Wind Power Forecasting by Using an Intelligent Statistical Treatment for Wind Resource Data. Energies. 2023;16(23):7915. https://doi.org/10.3390/en16237915

[9] Chang, Y., Yang, H., Chen, Y., Zhou, M., Yang, H., Wang, Y., et al. A Hybrid Model for Long-Term Wind Power Forecasting Utilizing NWP Subsequence Correction and Multi-Scale Deep Learning Regression Methods. IEEE Transactions on Sustainable Energy. 2024;15(1):263–275. https://doi.org/10.1109/tste.2023.3283242

[10] Shan, R., Niu, J., Chai, X., Gu, Q. Mid-to-long Term Wind Power Forecasting Based on ARIMA-BP Combined Model. Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering). 2024;17(4):401–407. https://doi.org/10.2174/2352096516666230818145947

[11] Ahmed, S. I., Ranganathan, P., Salehfar, H. Forecasting of Mid- and Long-Term Wind Power Using Machine Learning and Regression Models. 2021 IEEE Kansas Power and Energy Conference (KPEC); 2021. p. 1–6. https://doi.org/10.1109/kpec51835.2021.9446250

[12] Alkesaiberi, A., Harrou, F., Sun, Y. Efficient Wind Power Prediction Using Machine Learning Methods: A Comparative Study. Energies. 2022;15(7):2327. https://doi.org/10.3390/en15072327

[13] Ahmadi, A., Nabipour, M., Mohammadi-Ivatloo, B., Amani, A. M., Rho, S., Piran, Md. J. Long-Term Wind Power Forecasting Using Tree-Based Learning Algorithms. IEEE Access. 2020;8:151511–151522. https://doi.org/10.1109/access.2020.3017442

[14] Lin, W.-H., Wang, P., Chao, K.-M., Lin, H.-C., Yang, Z.-Y., Lai, Y.-H. Wind Power Forecasting with Deep Learning Networks: Time-Series Forecasting. Applied Sciences. 2021;11(21):10335. https://doi.org/10.3390/app112110335

[15] Wang, X., Liu, Y., Hou, J., Wang, S., Yao, H. Medium- and Long-Term Wind-Power Forecasts, Considering Regional Similarities. Atmosphere. 2023;14(3):430. https://doi.org/10.3390/atmos14030430

[16] Xiang, G., Lan, T., Liu, L. Medium and Long Term Wind Power Generation Forecasting Method Based on Multi-model Fusion. Journal of Physics: Conference Series. 2024;2774(1):012004. https://doi.org/10.1088/1742-6596/2774/1/012004

[17] Li, Y., Luo, X., Wu, W., Zhou, S., Zhang, Y., Zuo, Z. Low-rank Regularized Long-term Forecasting Model for Wind Power Forecasting. 2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC); 2024. p. 299–304. https://doi.org/10.1109/csis-iac63491.2024.10919277

[18] Tang, Y. S. Research on joint prediction of wind power, load and electricity price under multi-regional conditions [Master's thesis]. Guangdong University of Technology; 2025. https://doi.org/10.27029/d.cnki.ggdgu.2025.001579

[19] Tang, Q. W., Xiang, Y., Dai, J. K., Li, Z. H., Sun, W., Liu, J. Y. Transfer prediction method for wind farm output power based on CNN–LSTM. Advanced Engineering

Sciences. 2024;56(2):91–99. https://doi.org/10.15961/j.jsuese.202201165
[20] Xue, Y., Wang, L., Wang, S., Zhang, Y. F., Zhang, N. An ultra-short-term wind power forecasting model combining CNN and GRU networks. Renewable Energy Resources. 2019;37(3):456–462. https://doi.org/10.13941/j.cnki.21-1469/tk.2019.03.023
[21] Hua, Q., Fan, Z., Mu, W., Cui, J., Xing, R., Liu, H., et al. A Short-Term Power Load Forecasting Method Using CNN-GRU with an Attention Mechanism. Energies. 2025;18(1):106. https://doi.org/10.3390/en18010106
[22] Mockus, J., Mockus, L. Bayesian approach to global optimization and application to multiobjective and constrained problems. Journal of Optimization Theory and Applications. 1991;70(1):157–172.
[23] Deng, S. Hyper-parameter optimization of CNN based on improved Bayesian optimization algorithm. Application Research of Computers. 2019;36(7):1984–1988.
[24] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014;15(56):1929–1958.