electronics

MDPI

Article

A Robust Error Correction Strategy for High-Quality Day-Ahead Wind Power Interval Prediction in Transitional Weather

Bing Kang, Guili Ding *, Yixiong Xiao, Xiaocui Yuan, Zongyao Wang, Yongli Ma and Youxing Wei

School of Electrical Engineering, Jiangxi University of Water Resources and Electric Power, Nanchang 330099, China; kangbing@juwp.edu.cn (B.K.); 15907068315@163.com (Y.X.); yuanxc2012@163.com (X.Y.); 2010994239@juwp.edu.cn (Z.W.); marymayl@126.com (Y.M.); youxingwei@juwp.edu.cn (Y.W.)
* Correspondence: wyyx150051@163.com

Abstract

Weather transitions can cause rapid fluctuations in wind power and complicate the construction of prediction intervals. This study presents a robust error-correction framework that balances interval coverage and width. Multiple backward cloud transformation based on sampling with replacement (MBCT-SR) is combined with kernel-based fuzzy c-means (KFCM) clustering to represent and group error-related feature patterns. A temporal convolutional network and Transformer model (TCN-Transformer) predicts power and error sequences, and adaptive-bandwidth kernel density estimation (ABKDE) is used to construct prediction intervals. An improved multi-objective dung beetle optimizer (IMODBO) determines cluster-specific error-correction weights using coverage and width as competing objectives. The framework is evaluated using wind power data from China Longyuan Power Group Corporation Limited. The reported examples show reductions in interval width, with changes in coverage that depend on the test day and nominal coverage level.

Keywords: transitional weather; backward cloud transformation; error type clustering; correction weight optimization; kernel density estimation

1. Introduction

Wind energy contributes to the transition towards low-carbon power systems, but its variability creates challenges for forecasting and system operation [1,2]. Rapid changes in weather can produce abrupt changes in wind power output. Forecasting methods that account for these fluctuations are therefore needed to describe both the expected output and its uncertainty. This study investigates wind power interval forecasting during weather transitions.

Power forecasting includes physical and data-driven approaches [3]. Related methods have been reviewed for solar forecasting [4], while deep convolutional recurrent models have been investigated for wind power [5]. Wind studies have explored statistical ensembles [6], optimizer-tuned CNN-LSTM models [7], adversarial data generation for extreme weather [8], and wind-speed interval division for gale-weather forecasting [9]. Another study focuses on subsection forecasting during extreme weather [10]. These strategies address different modeling and data-processing tasks. Hybrid centroid- and graph-based methods have also been developed in general clustering research [11]. Weather-type clustering has been applied to hourly solar irradiance forecasting [12]. Information-gain feature selection and principal component analysis have been investigated for photovoltaic fault

Check for updates

Academic Editor: Fabio Corti

Received: 9 September 2026
Revised: 15 October 2026
Accepted: 19 October 2026
Published: 22 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

diagnosis [13]. These related applications motivate examining feature representations, while their value for wind power forecasting requires evaluation on the forecasting task.

Interval forecasts describe uncertainty around point predictions. Wind-energy studies have explored graph-based wind-speed interval prediction [14] and wind power frameworks using kernel density estimation (KDE) [15,16]. Related photovoltaic studies have considered scenario-based system operation under forecast uncertainty [17] and compared probabilistic day-ahead forecasting methods [18]. In the comparison reported in [18], conformalized quantile regression achieved the highest probabilistic accuracy among the tested approaches. Lower upper bound estimation (LUBE) directly estimates two interval bounds using a neural network trained with an objective combining coverage and width [19]. Criterion selection has also been discussed for wind power interval evaluation [20]. For shipboard solar irradiation, a graph-attention framework combines interval construction with prediction-error-based interval modification [21]. Bootstrap and quantile regression methods have been compared for quantifying photovoltaic point-forecast uncertainty [22]. Wind power point and interval forecasting has also been formulated as a multi-objective optimization problem [23]. These studies motivate evaluating interval coverage and width jointly.

Improving point forecasts and correcting residual errors are related approaches. A robust deep ensemble framework with error compensation has been investigated for load forecasting [24], while error correction has been combined with extreme learning machines for wind power prediction [25]. Solar forecasting studies have likewise considered dynamic error compensation [26] and data preprocessing and postprocessing [27]. A modified hidden Markov model has been proposed for wind-speed correction to improve wind power forecasting [28]. These applications concern different prediction targets and correction stages. Here, cluster-specific weights adjust predicted power errors, with interval coverage and width treated as competing objectives.

Recent frameworks combine fluctuation-type identification, cloud-based error modeling, and transfer learning [29] or transfer a point-prediction model into multi-objective interval learning [30]. Fu et al. [31] combine physically informed features with error-type encoding and multi-objective interval calibration in a twin-model framework. This study investigates cluster-specific correction of predicted power errors during weather transitions. A separate correction coefficient is assigned to each KFCM cluster, allowing for the adjustment to vary across error-related patterns. Interval coverage and normalized width are treated as competing objectives. The contribution is assessed through the reported forecasting and interval-correction comparisons, with their case-specific limitations. The main contributions are summarized below.

(1) MBCT-SR is used to fuse high-dimensional input features into cloud features for KFCM clustering of error-related patterns.

(2) ABKDE adjusts the bandwidth according to local data density to represent the prediction-error distribution.

(3) IMODBO optimizes cluster-specific error-correction weights using interval coverage and width as competing objectives.

Section 2.1 describes cloud-based feature fusion and KFCM clustering. Section 2.2 presents the forecasting, interval-construction, and error-correction components. Section 3 reports the case-study results, and Section 4 summarizes the findings and their limitations.

2. Materials and Methods

2.1. Feature Fusion Method for Transitional Weather Clustering

2.1.1. Principles of Backward Cloud Transformation Algorithm

The cloud model represents uncertainty through both randomness and fuzziness [32]. Its numerical characteristics are expectation E_X, entropy E_n, and hyperentropy H_e. Expectation describes the central location of cloud droplets. Entropy represents uncertainty in their distribution, and hyperentropy describes uncertainty in entropy. In this study, backward cloud transformation mapped quantitative input features to these three cloud characteristics.

Forward cloud generation maps qualitative cloud characteristics to quantitative droplets. Backward cloud transformation estimates those characteristics from quantitative samples. Here, MBCT-SR combined sample moments with repeated sampling with replacement to obtain a three-dimensional feature representation.

The method for calculating digital features in MBCT-SR is as follows:

(1) There are N cloud droplets X_i (i = 1,2,. . .,N), where N is the number of meteorological features at the sampling point. The equation for E_X is as follows:

$$E_X = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{1}$$

(2) Sample with replacement from the N cloud droplets, selecting r ($r \leq N$) cloud droplets each time. Repeat sampling m times to obtain the sample sets $\dot{X}_j = \left\{\dot{X}_{j1}, \dot{X}_{j2}, \ldots, \dot{X}_{jr}\right\}, (j = 1, 2, \ldots, m)$. The equations for calculating the mean $\overline{\dot{X}_j}$ and variance Y_j^2 of $\dot{X}_j$ are as follows:

$$\overline{\dot{X}_j} = \frac{1}{r}\sum_{k=1}^{r} \dot{X}_{jk} \tag{2}$$

$$Y_j^2 = \frac{1}{r-1}\sum_{k=1}^{r} \left(\dot{X}_{jk} - \overline{\dot{X}_j}\right)^2 \tag{3}$$

(3) E_n and H_e are calculated indirectly from $Y^2 = \left\{Y_1^2, Y_2^2, \ldots, Y_m^2\right\}$. First, the mean E_{Y^2} and variance D_{Y^2} of Y^2 are calculated from Y^2:

$$E_{Y^2} = \frac{1}{m}\sum_{j=1}^{m} Y_j^2 \tag{4}$$

$$D_{Y^2} = \frac{1}{m-1}\sum_{j=1}^{m} \left(Y_j^2 - E_{Y^2}\right)^2 \tag{5}$$

(4) The calculation equations for E_n and H_e are as follows:

$$E_n = \sqrt{\frac{\sqrt{4(E_{Y^2})^2 - 2D_{Y^2}}}{2}} \tag{6}$$

$$H_e = \sqrt{E_{Y^2} - (E_n)^2} \tag{7}$$

MBCT-SR uses resampling to estimate cloud features from high-dimensional inputs. The resulting three-dimensional representation is then used for clustering, as described below.

2.1.2. Principles of KFCM Clustering Algorithm

KFCM introduces a kernel function into fuzzy c-means clustering to represent nonlinear relationships among samples. In this study, a Gaussian kernel is used to group the cloud features obtained by MBCT-SR. The implementation steps are described below.

(1) Define the basic parameters of KFCM: Gaussian kernel function $K(X_j,v_i)$ parameters δ, objective function $J_m(U,V)$, number of clusters c, number of samples n, convergence accuracy ε, fuzzy weighted index m, maximum number of iterations k. The equations for $K(X_j,v_i)$ and $J_m(U,V)$ are as follows:

$$K(X_j, v_i) = exp\left\{-\frac{\left\|X_j - v_i\right\|^2}{2\delta^2}\right\} \tag{8}$$

$$J_m(U, V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m (2 - 2K(v_i, X_j)) \tag{9}$$

Here, U is the membership matrix, and V is the cluster center matrix.

(2) Calculate the membership u_{ij} of sample X_j in the ith cluster and the cluster center v_i of the ith cluster using the following equations:

$$u_{ij} = \frac{(1 - K(X_j, v_i))^{\frac{1}{m-1}}}{\sum_{g=1}^{c}(1 - K(X_j, v_g))^{\frac{1}{m-1}}} \tag{10}$$

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m K(X_j, v_i) X_j}{\sum_{j=1}^{n} u_{ij}^m K(X_j, v_i)} \tag{11}$$

(3) Update U and V at each iteration until the following termination condition is met or the number of iterations t reaches the maximum k. Return the clustering result:

$$\left\|U^{(t)} - U^{(t-1)}\right\| < \varepsilon \tag{12}$$

Internal indicators are unsupervised, do not require a benchmark data set, and do not rely on external reference models. Clustering is evaluated using the silhouette coefficient (SC) and the sum of squared errors (SSE). These measures are calculated as follows:

$$SC = \frac{1}{n}\sum_{j=1}^{n}\frac{q(j) - p(j)}{max[p(j), q(j)]} \tag{13}$$

$$SSE = \sum_{i,j}(X_{ij} - V_i)^2 \tag{14}$$

p is the average distance between a sample and other samples in the cluster where the sample belongs, and q is the average distance between a sample and all samples in other clusters. The value range of SC is $[-1,1]$; the closer the value is to 1, the better the clustering effect is. The smaller SSE is, the closer the data is to their respective cluster centers, and the better the clustering effect is.

2.2. Robust Error Correction Method

The framework combines power and error-sequence forecasting with cluster-specific correction and interval construction. TCN-Transformer models first predict power and its error sequence. For each candidate error-correction weight vector, the weighted predicted errors adjust the power forecast and ABKDE constructs the resulting prediction intervals.

IMODBO evaluates their coverage and width as competing objectives and searches for a compromise weight vector.

2.2.1. Power and Error Forecasting with a TCN-Transformer Model

The predictor uses convolutional encoding and self-attention to extract information from the input sequence. Dilated causal convolutions and residual connections encode temporal patterns. Positional embeddings are added before the self-attention layers, and the representation selected at the last sequence position is passed to a fully connected output layer. The feature representation contains multiple components; it is not obtained by replacing all nine original features with one scalar.

The TCN component uses dilated causal convolutions to capture temporal patterns over an expanded receptive field. Its dilation factor increases exponentially with network depth. Residual connections support gradient propagation between adjacent layers. Each residual block contains two convolutional layers and nonlinear transformations, with normalization and dropout for regularization. The Transformer component uses self-attention to model dependencies, and multi-head attention allows for parallel computation. Figure 1 provides a schematic overview of the TCN-Transformer model.

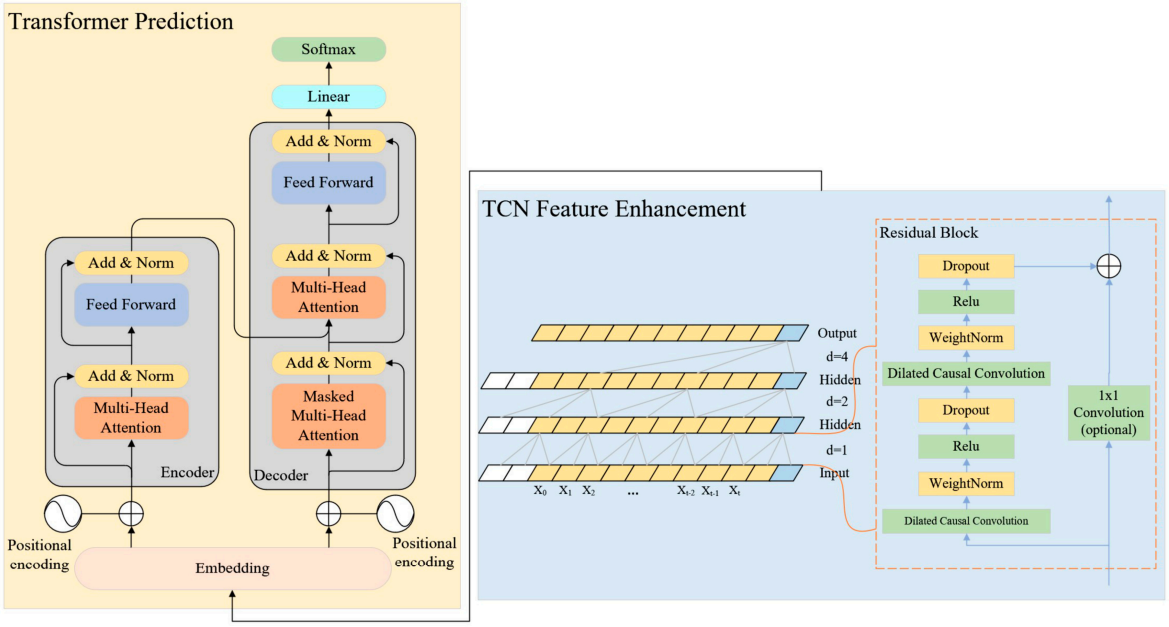


Figure 1. A schematic overview of the TCN-Transformer model. The pale-yellow and pale-blue panels denote the Transformer prediction and TCN feature-enhancement modules, respectively.

2.2.2. Prediction-Interval Construction Using Adaptive-Bandwidth Kernel Density Estimation

Fixed-bandwidth kernel density estimation uses a single bandwidth across the sample distribution. In this study, ABKDE started from a fixed bandwidth and adapted it to local data density. Smaller bandwidths were used in dense regions and larger bandwidths in sparse regions. Prediction intervals derived from the estimated density were evaluated using both coverage and width, as reported in Section 3.3.

ABKDE first needs to determine the optimal fixed bandwidth and use it as the initial bandwidth to continuously optimize globally. The optimal fixed initial bandwidth is calculated according to the asymptotic mean integrated square error criterion *AMISE*. The choice of kernel function has little effect on the result of kernel density estimation, so

the Gaussian kernel function with the best robustness was selected in this paper, and the calculation equation is as follows:

$$AMISE(h) = \frac{h^4[\int x^2 G(x)dx]^2 \int [f''(x)]^2 dx}{4} + \frac{\int [G(x)]^2 dx}{Dh} \quad (15)$$

Here, h is the bandwidth. $G(x)$ is the Gaussian kernel function; $f''(x)$ is the second derivative of the probability density function $f(x)$, and D is the number of sample points x. The equation for $f(x)$ is as follows:

$$f(x) = \frac{1}{Dh} \sum_{d=1}^{D} G(\frac{x - x_d}{h}) \quad (16)$$

Setting the derivative of *AMISE* to zero gives the optimal fixed bandwidth h_{AMISE}:

$$h_{AMISE} = \frac{\left\{\int [G(x)]^2 dx\right\}^{\frac{1}{5}}}{[\int x^2 G(x)dx]^{\frac{2}{5}} \left\{\int [f''(x)]^2 dx\right\}^{\frac{1}{5}} D^{\frac{1}{5}}} \quad (17)$$

Substitute h_{AMISE} into Equation (16) to obtain the preliminary fixed-bandwidth density estimate $f_{AMISE}(x)$.

It can be seen from (15) that each sample point is related to variance and bias. Therefore, in the case of complex sample data distribution, fixed bandwidth kernel density estimation will inevitably have the problem of poor local adaptability, causing the estimated results to deviate from the actual distribution. Therefore, this paper used the optimal fixed bandwidth h_{AMISE} as the initial bandwidth and added the sensitivity factor α to achieve adaptive adjustment of the bandwidth according to the density of data in the global range. The calculation equation of $h(x_d)$ is as follows:

$$h(x_d) = \left\{ \frac{[\prod_{d=1}^{D} f_{AMISE}(x_d)]^{\frac{1}{D}}}{f(x_d)} \right\}^{\alpha} \quad (18)$$

Taking *AMISE* as the objective function, the sensitivity factor α is optimized globally through the simple golden section method to obtain the optimal bandwidth of each point, and finally, the prediction interval is constructed.

2.2.3. Cluster-Specific Error Correction Using an Improved Multi-Objective Dung Beetle Optimizer

Residuals are defined as measured minus predicted power. Directly adding a predicted residual can overcorrect or undercorrect local errors. Meteorological feature patterns are therefore grouped into error-related categories, and each predicted residual is multiplied by the coefficient for its category before being added to the power forecast. The coefficients control the contribution of the residual forecast rather than replacing the original power forecast.

IMODBO selects one coefficient for each error-pattern cluster, with each coefficient constrained to [0,2]. For each candidate vector, the weighted predicted residuals correct the power forecast and ABKDE constructs the corresponding prediction intervals. The two objectives are the uncovered proportion, 1−PICP, and the prediction interval normalized average width, PINAW. IMODBO generates a Pareto set, from which a compromise coefficient vector is selected. Among solutions of comparable width, selection prioritizes

coverage. The selected weights are then used for cluster-specific correction and ABKDE interval construction. The search strategies are described below.

Dung beetle optimization (DBO), proposed by Xue and Shen [33], is a swarm-intelligence algorithm for single-objective optimization. This study adapted DBO to the competing objectives of interval coverage and width and introduced the population-management and search-update strategies described below.

Dung beetle populations have five behaviors: rolling, dancing, spawning, foraging, and stealing. First, the population is initialized to generate individual B_i. Then, the dynamic quasi-opposition-based learning strategy is used to generate new individuals B_R and update the population. Dynamic quasi-opposition-based learning further improves randomness on the basis of opposition-based learning and effectively improves population diversity. At the same time, the variation range of the opposition solution can be adjusted according to the fitness to avoid generating too many invalid individuals. The calculation equation is as follows:

$$B_R = \begin{cases} \frac{f_1(B_z)+f_2(B_z)}{f_1^{best}+f_2^{best}} R\left(\frac{B_{max}+B_{min}}{2}, B_{max}+B_{min}-B_z\right) \\ \frac{f_1(B_z)+f_2(B_z)}{f_1^{best}+f_2^{best}} R\left(B_{max}+B_{min}-B_z, \frac{B_{max}+B_{min}}{2}\right) \end{cases} \tag{19}$$

where B_z is the ith individual, B_{max} and B_{min} are the upper and lower bounds of the individual, and $R()$ represents a random number within the range; f_1 and f_2 are two objective functions. f_1^{best} and f_2^{best} are the optimal solutions of f_1 and f_2 in the current population. In order to clearly show the Pareto front of the solution set, f_1 is set to the interval uncovered rate, which is numerically equal to 1 minus interval coverage rate, and f_2 is set to interval width. The calculation equations are as follows:

$$f_1 = 1 - \frac{1}{D}\sum_{d=1}^{D} C_d \tag{20}$$

$$f_2 = \frac{1}{DQ}\sum_{d=1}^{D} (U_d - L_d) \tag{21}$$

When the actual value is within the interval [L_d,U_d], C_d = 1, otherwise C_d = 0. Q is the range of the predicted value, and U_d and L_d are the upper and lower bounds of the interval, respectively.

Then periodic cyclic population reduction strategy is designed to accelerate the convergence speed and maintain the diversity of the population. Some individuals are selected to be removed from the population according to the current number of iterations to speed up convergence. After reaching the minimum population, a cycle is completed. Then, the removed individuals are moved into the population to restore the population to the maximum to improve diversity and avoid falling into the local optimum, and then, the next cycle begins. The calculation equation is as follows:

$$N(t+1) = N_{min} + (N - N_{min}) \times \left[1 - \left(\frac{t\%\frac{T_{max}}{T}}{\frac{T_{max}}{T}}\right)\right] \tag{22}$$

Here, N is the current population size. N_{min} is the minimum population limit; t is the current iteration number. T_{max} is the maximum number of iterations, and % is the modulo operation. After initializing the population each time, the fitness value of each individual is calculated, and the fitness is sorted. The non-dominated solutions are identified, and they are stored in the file.

The rolling behavior of the dung beetle is affected by the intensity of the light source ΔB, and the position update equation is as follows:

$$\begin{pmatrix} B_z(t+1)=B_z(t)+\beta\times k\times\Delta B_z(t-1)+b\times\Delta B \\ \Delta B=|B_z(t)-B^w| \end{pmatrix} \tag{23}$$

Here, $B_z(t)$ is the position information of the ith individual at the tth iteration. B^w is the global worst position. β is the natural coefficient; k is the deflection coefficient, and b is a random number.

Dung beetles dance to reorient themselves when they encounter obstacles. The position update equation is as follows:

$$B_z(t+1) = B_z(t) + tan(\theta)|B_z(t) - B_z(t-1)| \tag{24}$$

Here, θ is the deflection angle in [0,π].

Female dung beetles choose a suitable place to lay their eggs. The position update equations are as follows:

$$\begin{pmatrix} LB^*=max(B^*\times(1-Y),B_{min}) \\ UB^*=max(B^*\times(1+Y),B_{max}) \end{pmatrix} \tag{25}$$

$$B_z(t+1) = B^* + b_1 \times (B_z(t) - LB^*) + b_2 \times (B_z(t) - UB^*) \tag{26}$$

Here, LB^* and UB^* are the lower and upper bounds of the spawning area, $Y = 1 - t/T_{max}$, and B^* is the current local optimal position; b_1 and b_2 are two independent random vectors.

Adult dung beetles forage for food. The position update equations are as follows:

$$\begin{pmatrix} LB^b=max(B^b\times(1-Y),B_{min}) \\ UB^b=max(B^b\times(1+Y),B_{max}) \end{pmatrix} \tag{27}$$

$$B_z(t+1) = B^b + c_1 \times (B_z(t) - LB^b) + c_2 \times (B_z(t) - UB^b) \tag{28}$$

Here, LB^b and UB^b are the lower and upper bounds of the foraging area, and B^b is the global optimal position. c_1 is a normally distributed random number, and c_2 is a random vector.

Some dung beetles steal balls from other dung beetles. The position update equation is as follows:

$$B_z(t+1) = B^b + M \times g \times (|B_z(t) - B^*| + \left|B_z(t) - B^b\right|) \tag{29}$$

Here, M is a constant value, and g is a random vector.

When the algorithm is close to convergence in the later iteration, it is necessary to avoid the continuous opposite change of the search direction. This paper designed an interpolation smoothing strategy to improve the convergence speed and accuracy of the algorithm. The calculation equation is as follows:

$$\begin{pmatrix} B_z(t)=\lambda B_{z-1}(t)+(1-\lambda)B_{z+1}(t) \\ if\ sign(B_z(t)-B_{z-1}(t))sign(B_{z+1}(t)-B_z(t))<0 \end{pmatrix} \tag{30}$$

Here, λ is a random number, and *sign* denotes the sign function.

The algorithm returns the stored Pareto solution set after reaching the maximum number of iterations. Figure 2 summarizes the IMODBO procedure.

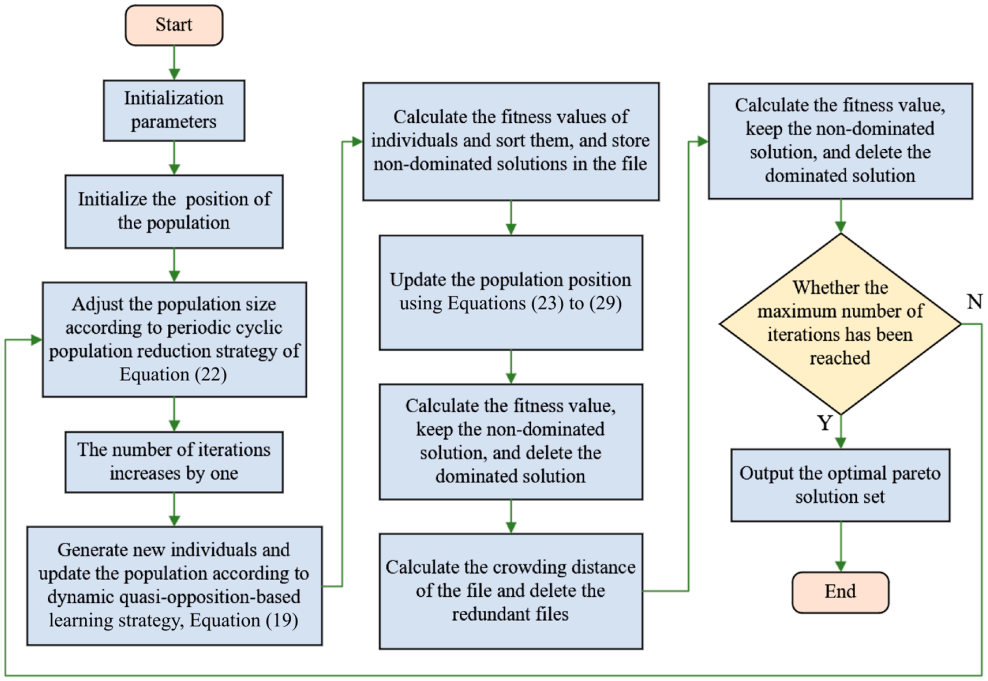


Figure 2. Flowchart of IMODBO.

3. Case Study

This case study uses records collected during 2019–2020 at a wind farm operated by China Longyuan Power Group Corporation Limited in western China. The prediction object is a single wind turbine with a rated capacity of 1523 kW, and the reported sampling interval is 15 min. The supplied source feature table contains 10,001 numeric records, and the processed working table contains 6888 records. These counts describe the retained source and working records; supervised-window counts depend on input construction. The forecasting target Patv is active power measured in kW, while Prtv is a reactive-power input. All methods are evaluated using MATLAB R2023b. Table 1 lists the original input features.

Table 1. High-dimensional features of the original dataset.

Feature Name	Specification
Wspd (m/s)	The wind speed recorded by the anemometer
Wdir (°)	The angle between the wind direction and the position of turbine nacelle
Etmp (°C)	Temperature of the surrounding environment
Itmp (°C)	Temperature inside the turbine nacelle
Ndir (°)	Nacelle direction
Pab1 (°)	Pitch angle of blade 1
Pab2 (°)	Pitch angle of blade 2
Pab3 (°)	Pitch angle of blade 3
Prtv	Reactive power

Transitional-weather cases are selected from the intersection of weather periods and historical power-transition periods. Historical power is smoothed using a two-sample moving average applied once. Local extrema define inflection points, and intervals between adjacent points form candidate periods. The main power-transition criterion uses a capacity-relative threshold factor of 0.15. Periods that fail the initial criterion receive a secondary within-period fluctuation check with a coefficient of 0.5 relative to their maximum power. Both decision branches require a duration exceeding 1 h, and adjacent transition periods are

merged. SO-XGBoost first classifies icing, high-temperature, heavy-precipitation, and other conditions. Wind-speed fluctuation periods are extracted only from the other category, using normalized lower and peak thresholds of 0.3 and 0.6. A fluctuation process starts at the lower threshold, includes peaks meeting the 0.6 rule, and returns to the lower threshold. The classifier uses a 7:3 training/test split; SO uses 50 individuals and 50 iterations to optimize tree number, tree depth, and learning rate using classification error. Weather and power-transition periods are matched by their common sampling points. Weather labels are distinct from the KFCM error-pattern clusters. Figure 3 summarizes the workflow.

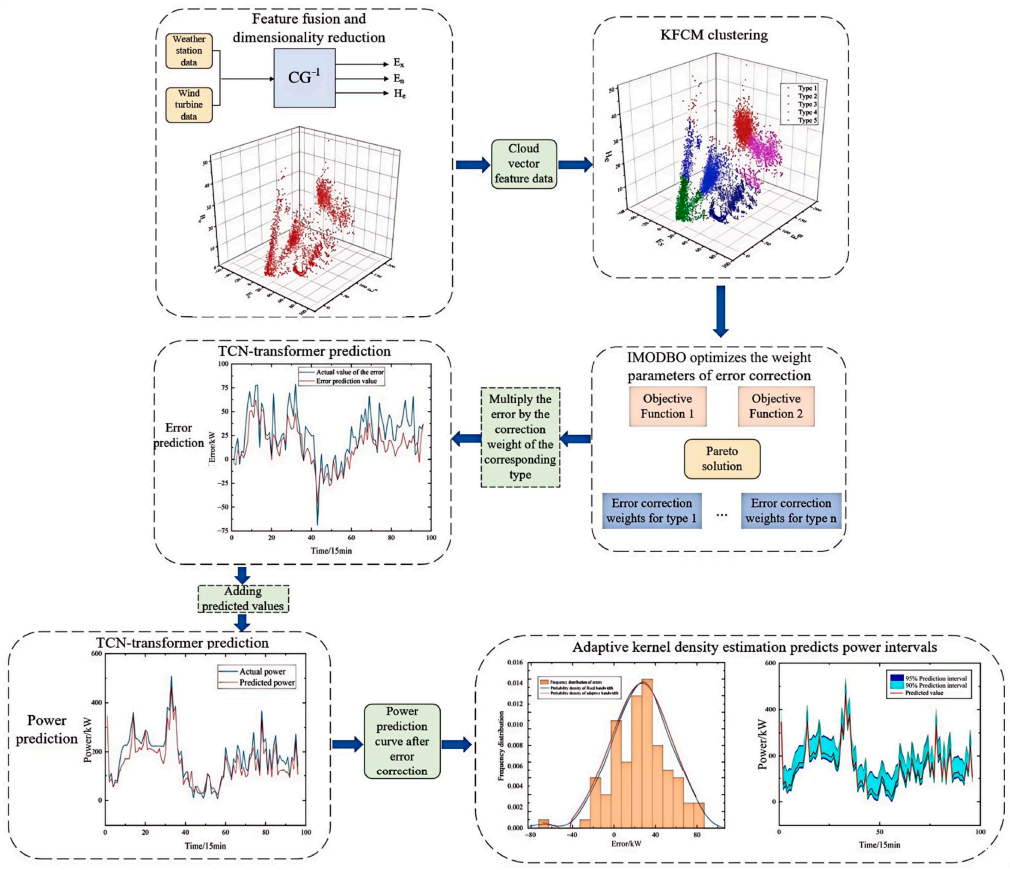


Figure 3. Complete experimental process.

3.1. Clustering Result Analysis

The dataset contains nine original features. The reported MBCT-SR configuration uses 100 resampling operations, with seven features selected per operation. KFCM is compared with Gaussian mixture modeling (GMM), Kmeans, Kmeans++, and self-organizing maps (SOM), using a maximum of 100 iterations. Table 2 reports *SC* and *SSE* for three to six cluster centers.

Table 2 shows that the ranking depends on the chosen criterion and number of clusters; a higher *SC*, for example, need not coincide with lower within-cluster dispersion. With five clusters, KFCM has an *SC* of 0.626, compared with 0.637 for Kmeans++ with six clusters, and its *SSE* is 1.1153, compared with 3.400. These values support a joint assessment of separation and compactness rather than a single ranking that applies to every criterion and cluster count.

For comparison with MBCT-SR, principal component analysis (PCA) is applied to the original features, and the first three principal components are used for clustering with the same set of algorithms. Table 3 reports the clustering metrics for this representation.

Table 2. Comparison of clustering results based on MBCT-SR feature dimensionality reduction.

Cluster Centers		3	4	5	6
GMM	*SC*	0.515	0.556	0.547	0.456
	SSE (10^6)	6.914	5.330	4.809	4.25
Kmeans	*SC*	0.574	0.565	0.625	0.615
	SSE (10^6)	7.434	4.875	3.624	2.834
Kmeans++	*SC*	0.639	0.623	0.627	0.637
	SSE (10^6)	6.117	4.652	3.615	3.400
SOM	*SC*	0.544	0.451	0.588	0.573
	SSE (10^6)	11.732	9.617	10.373	5.885
KFCM	*SC*	0.622	0.609	0.626	0.597
	SSE (10^6)	8.9065	6.9046	1.1153	8.902

Table 3. Comparison of clustering results based on PCA feature dimensionality reduction.

Cluster Centers		3	4	5	6
GMM	*SC*	0.220	0.292	0.319	0.254
	SSE (10^6)	27.323	18.835	13.955	13.649
Kmeans	*SC*	0.54	0.529	0.584	0.619
	SSE (10^6)	12.993	10.969	8.290	6.356
Kmeans++	*SC*	0.512	0.541	0.554	0.616
	SSE (10^6)	14.459	10.545	8.534	6.354
SOM	*SC*	0.464	0.477	0.520	0.488
	SSE (10^6)	22.669	16.327	12.616	11.824
KFCM	*SC*	0.477	0.515	0.563	0.636
	SSE (10^6)	3.367	3.366	3.365	3.364

Table 3 reports that KFCM with six clusters has an *SC* of 0.636 and an *SSE* of 3.364. Tables 2 and 3 summarize clustering in the cloud and PCA representations, respectively. The *SC* and *SSE* values describe different aspects of those representations; they do not by themselves establish preservation of all forecasting-relevant information. In particular, comparing squared distances across representations depends on their coordinate scaling and requires matched preprocessing. Figure 4 shows the MBCT-SR-based clustering results.

3.2. Power and Error-Sequence Forecasting Results

The first 80% of the data is assigned to model development and correction calibration, while the final 20% is reserved for post-training evaluation. Within the development portion, the correction stage uses an 8:2 fitting/internal-evaluation split. Clustering, power and residual-model learning, correction-weight selection, and ABKDE fitting are confined to the development portion under this protocol. The historical input window is four days. At a 15 min sampling interval, a continuous four-day window contains 384 time points, with nine original features per point. The forecasting horizon is the following 24 h, corresponding to 96 points on the same grid. During recursive prediction, corrected power forecasts are used in subsequent power inputs. When the rolling window advances by one day, the retained three-day history is combined with the forecasted power segment. Only measurements or meteorological forecasts available at the forecast origin are admissible inputs; future measured SCADA values are not assumed available. This availability rule separates the day-ahead target from hindsight information. Table 4 lists the TCN-Transformer settings.

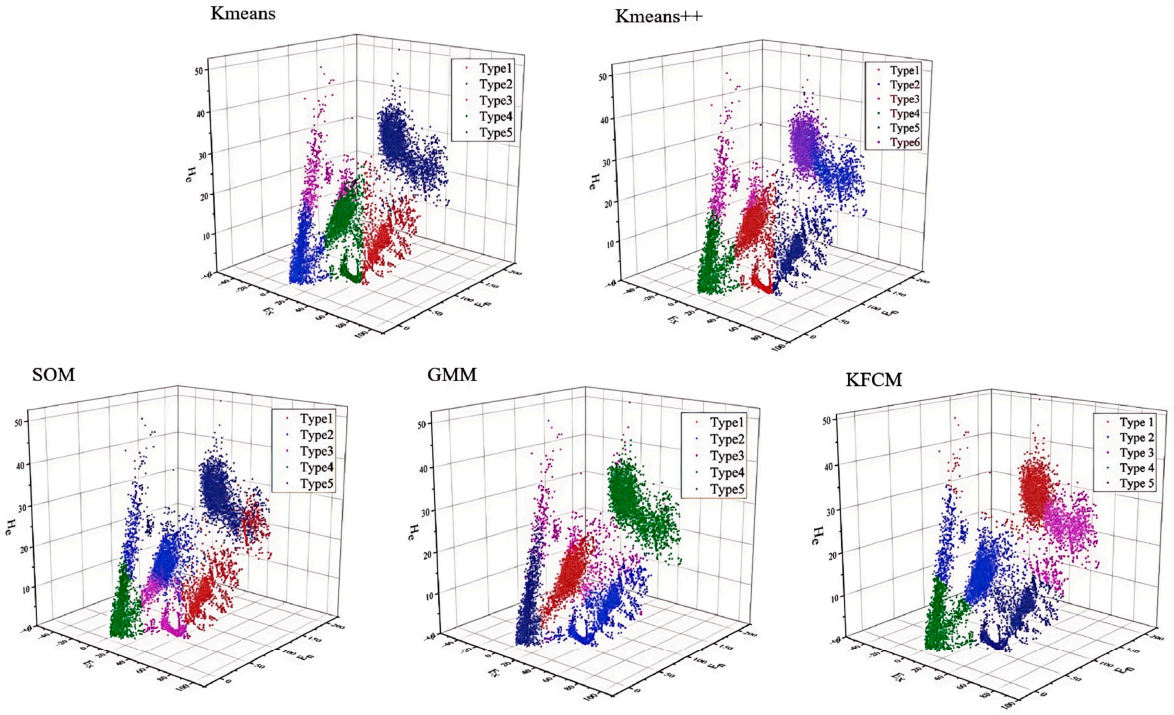


Figure 4. Clustering results of five algorithms.

Table 4. Reported parameter settings of the TCN-Transformer model.

Parameter Name	Parameter Value
Activation function	Relu
Optimizer	Adam
Learning Rate	{0.001, 0.0005, 0.00025}
Dropout	10%
Num channels	32
Max epochs	100
kernel size	3
Historical input window (days)	4 (384 points)
Num attention heads	2

The TCN-Transformer model is compared with long short-term memory (LSTM), support vector machine (SVM), and extreme learning machine (ELM) models. Results for three illustrated test days are reported in Table 5. Figures 5 and 6 compare the power and error-sequence predictions, respectively.

Table 5. A comparison of point-prediction errors for three illustrated test days. RMSE is reported in kW and MAPE as a ratio.

Model		Day1	Day2	Day3
LSTM	RMSE	64.135	76.157	88.38
	MAPE	0.065	0.171	0.129
SVM	RMSE	112.881	40.875	89.613
	MAPE	0.104	0.086	0.139
ELM	RMSE	115.683	53.068	93.283
	MAPE	0.102	0.127	0.141
TCN-Transformer	RMSE	46.9356	53.622	93.108
	MAPE	0.043	0.077	0.127

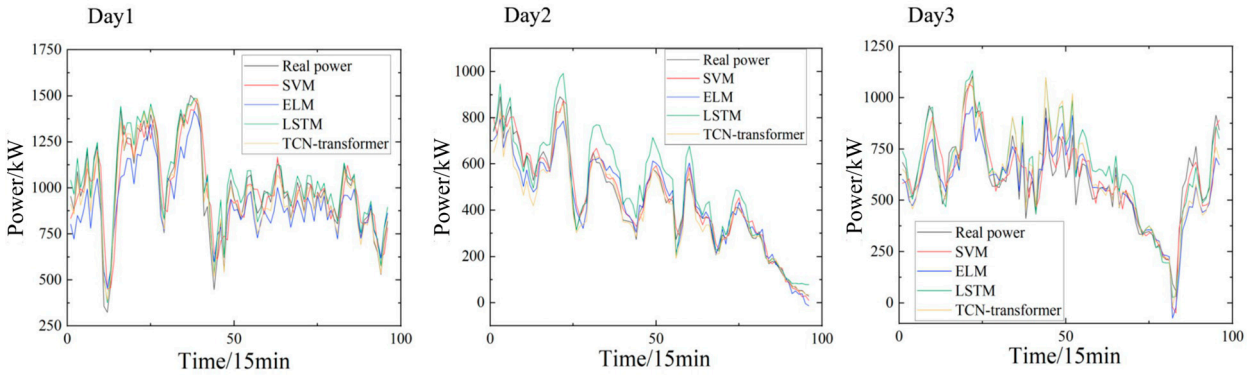


Figure 5. Comparison of power prediction results for three days.

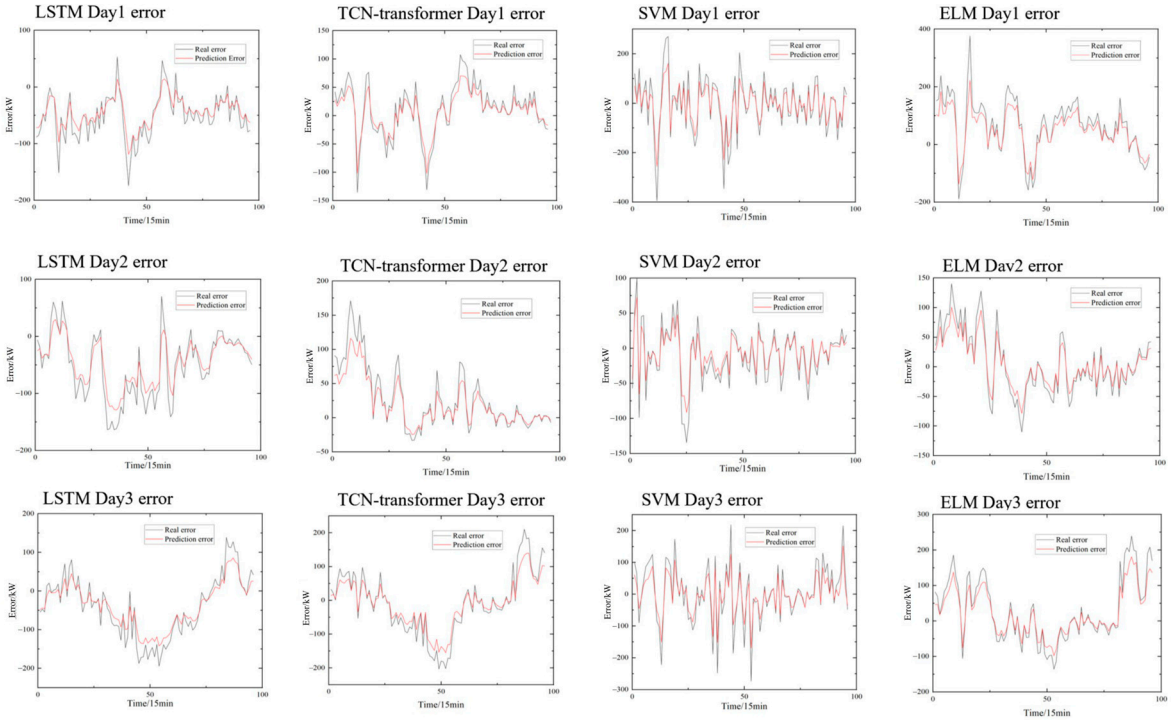


Figure 6. Comparison of error prediction results for three days.

Table 5 shows that TCN-Transformer has the lowest MAPE on all three illustrated days. Its RMSE is lowest on Day 1 (46.9356), whereas SVM has the lowest RMSE on Day 2 (40.875) and LSTM has the lowest RMSE on Day 3 (88.38). Thus, the ranking depends on the error metric and test day. Figure 7 shows the corresponding error distributions.

Figure 7 shows a relatively narrow error distribution for TCN-Transformer across the illustrated predictions. This aggregate distribution should be interpreted alongside the day-specific RMSE and MAPE values in Table 5.

3.3. Prediction-Interval Results and Error Correction

The same three illustrated test days are used to compare ABKDE with fixed-bandwidth kernel density estimation (KDE) at nominal coverage levels of 90% and 95%. Figure 8 presents the prediction intervals. Figure 9 compares the prediction-error density estimates, and Table 6 reports the interval-quality metrics.

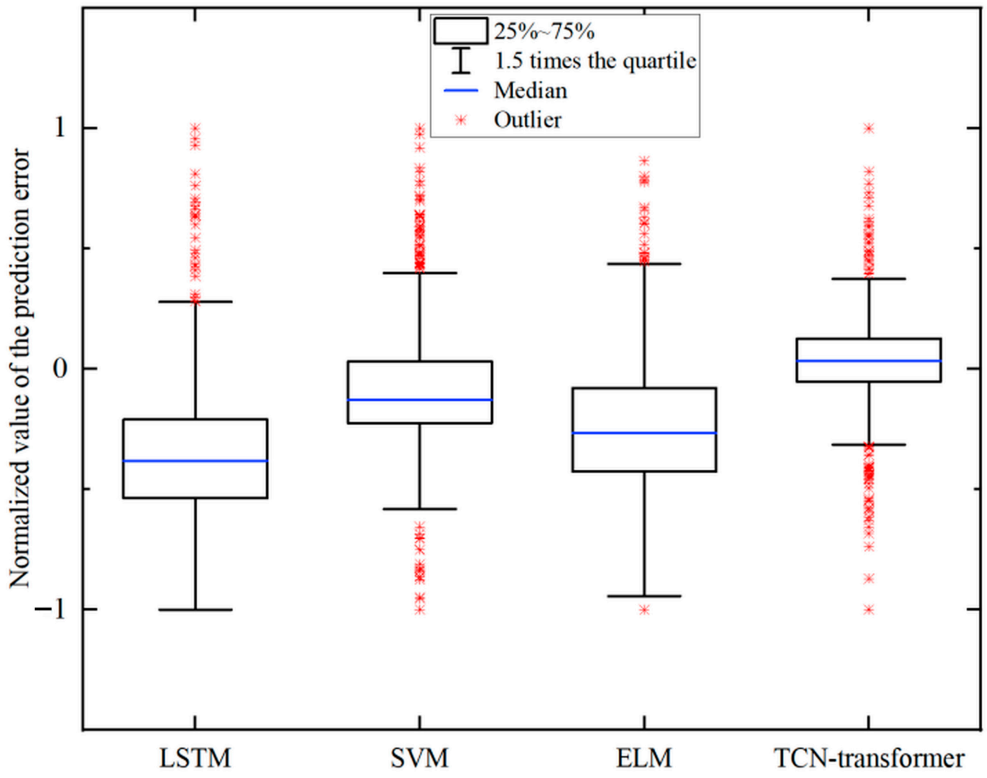


Figure 7. Error distribution box plots of four models.

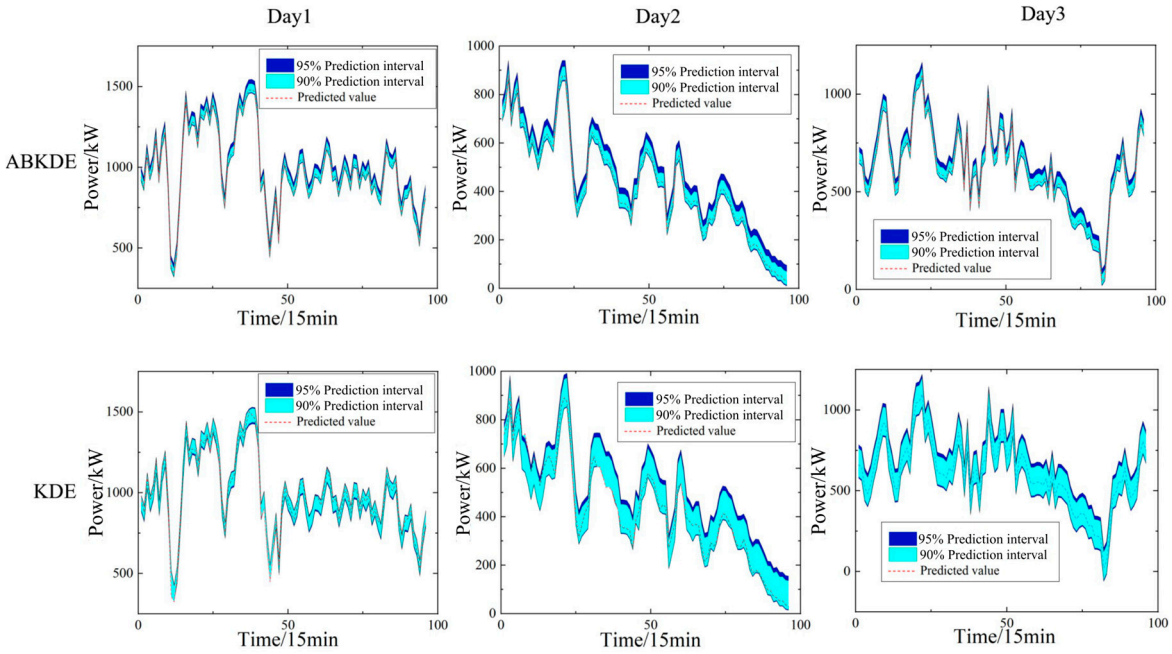


Figure 8. Prediction intervals obtained using KDE and ABKDE.

Table 6 shows that the coverage-width trade-off depends on the day and nominal coverage level. On Day 2 at 90%, ABKDE increases PICP from 0.8958 to 0.9688 and reduces PINAW from 0.0692 to 0.0649. On Day 3 at 90%, PICP increases from 0.7396 to 0.8229, while PINAW increases from 0.0649 to 0.0704. The latter PICP remains below the nominal level. These results indicate that coverage and width must be evaluated jointly rather than treating a change in either metric as a uniform improvement.

The correction-weight constraint interval is [0,2]. The reported IMODBO search uses a population size of 20, a maximum of 50 iterations, and an external repository of 20 solutions. Figure 10 shows the reported Pareto solutions, and Figure 11 shows the resulting corrected prediction intervals.

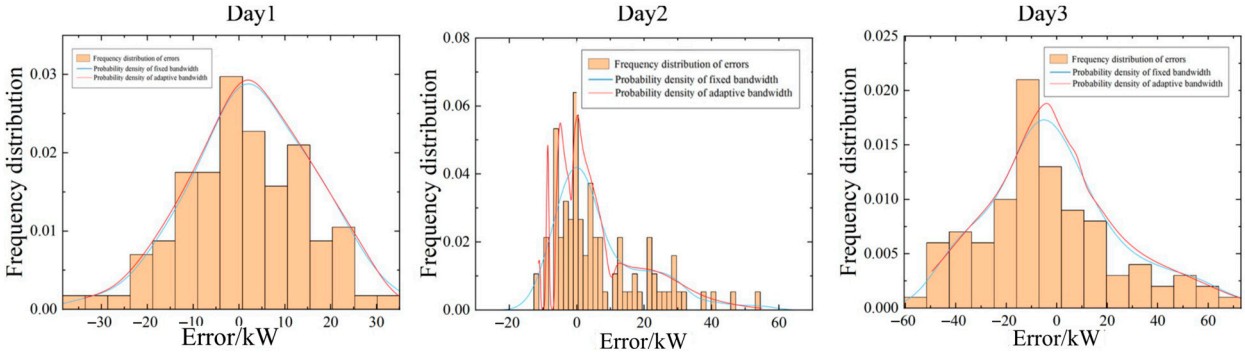


Figure 9. Comparison of prediction-error density estimates.

Table 6. Prediction-interval metrics for KDE and ABKDE at different nominal coverage levels.

Method	Confidence	Index	Day1	Day2	Day3
KDE	95%	PICP	1	0.9792	0.7917
		PINAW	0.069	0.0942	0.0823
	90%	PICP	0.9688	0.8958	0.7396
		PINAW	0.0507	0.0692	0.0649
ABKDE	95%	PICP	0.9583	1	0.8854
		PINAW	0.0663	0.0998	0.0814
	90%	PICP	0.9375	0.9688	0.8229
		PINAW	0.0475	0.0649	0.0704

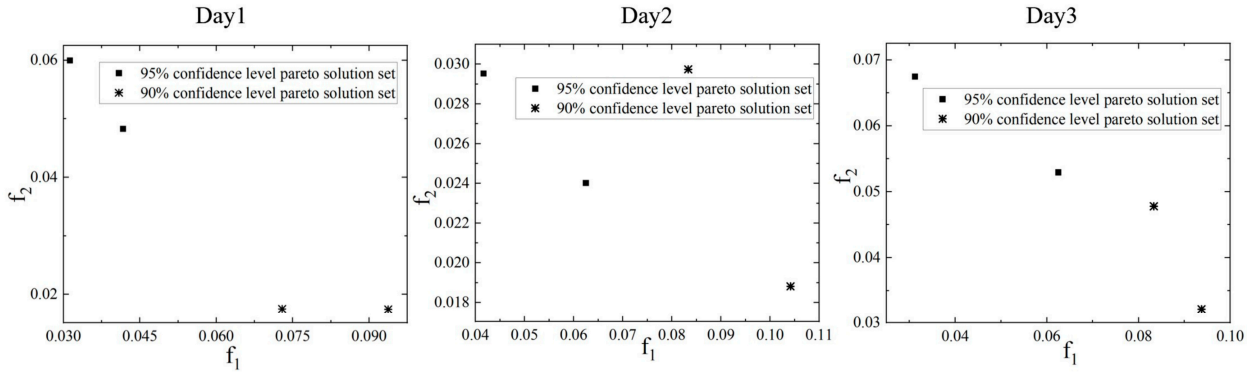


Figure 10. Pareto solutions for three days at different confidence levels.

The reported run takes 447.1553 s, with an average iteration time of about 9 s. Two five-dimensional correction-weight vectors are reported, [1.6115,0.5082,1.5309,0.9917,0.2645] and [0.6666,0.4007,0.6821,0.4678,1.4539]. The first vector is selected for the reported evaluation, giving priority to coverage. The interval metrics are reported in Table 7.

Correction calibration follows the development/evaluation allocation in Section 3.2. Power and residual predictors are learned using development records, and meteorological feature patterns determine the error categories. For each candidate coefficient vector, predicted residuals are weighted by category and added to the power forecasts. ABKDE fits the corresponding residual density and constructs prediction intervals; IMODBO scores

the candidates using 1−PICP and PINAW on the internal development evaluation subset. A compromise vector is selected from the Pareto solutions. The selected correction and density parameters are then fixed for evaluation on the final 20%, which is not used to select them.

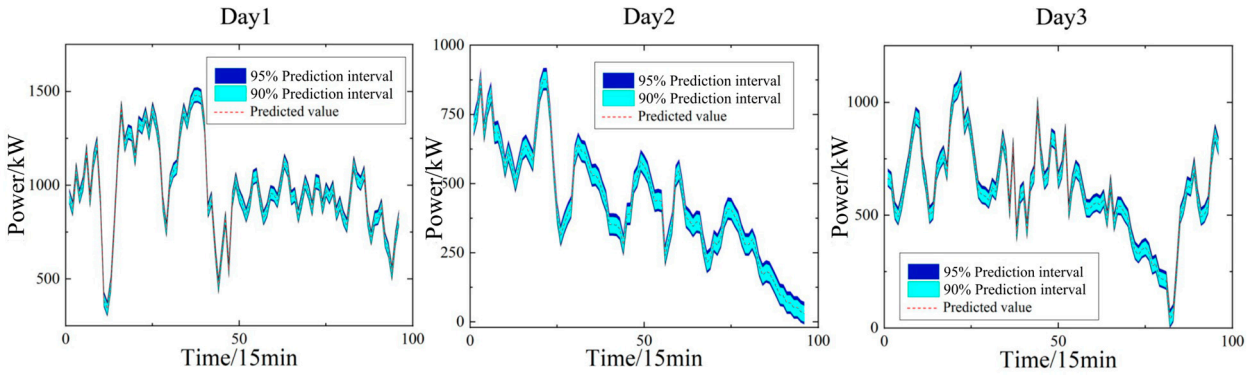


Figure 11. Prediction intervals after robust error correction.

Table 7. Prediction-interval metrics after cluster-specific error correction using IMODBO and ABKDE.

	Confidence	Index	Day1	Day2	Day3
ABKDE	95%	PICP	0.9688	0.9583	0.9375
		PINAW	0.0599	0.0295	0.0529
	90%	PICP	0.9271	0.9167	0.9063
		PINAW	0.0175	0.0297	0.0322

Table 7 reports lower PINAW than the ABKDE results in Table 6 in all six illustrated cases, but coverage does not increase in every case. On Day 2 at 95%, PINAW decreases from 0.0998 to 0.0295, while PICP decreases from 1.0000 to 0.9583. On Day 3 at 95%, PICP increases from 0.8854 to 0.9375 but remains below the nominal coverage level, and PINAW decreases from 0.0814 to 0.0529. These results show that the correction changes the coverage-width trade-off and that narrower intervals do not necessarily achieve the nominal coverage.

IMODBO is compared with MODBO and multi-objective particle swarm optimization (MOPSO) on four representative bi-objective test functions. For this benchmark comparison, all algorithms use a population size of 200, a maximum of 100 iterations, and an external repository of 200 solutions. Table 8 reports the results.

The evaluation indicators selected in this paper are the convergence index, inverted generational distance (IGD); the distribution index, Spacing; and the comprehensive performance index, Hypervolume (HV), where IGD represents the average distance from the points on the real pareto front to the points in the algorithm solution set. The smaller IGD is, the better the algorithm performance is. Spacing measures the uniformity of the distribution of the solution set on the pareto front. The smaller Spacing is, the denser the solution is, and the higher the diversity is. HV measures the size of the target space covered by the solution set generated by the algorithm. The larger HV is, the better the algorithm performance is.

Table 8 reports lower IGD and Spacing, higher HV, and shorter running times for IMODBO than for MODBO and MOPSO on the four benchmark functions. Using the rounded values reported in the table, the largest reductions relative to MODBO are 17.39% for IGD, 38.71% for Spacing, and 45.96% for running time; the largest HV increase is approximately 0.16%. These reported comparisons do not by themselves quantify variability across independent runs or establish performance against other multi-objective optimizers.

Table 8. Reported performance metrics of three multi-objective algorithms on the benchmark functions.

Test Function	Index	IMODBO	MODBO	MOPSO
ZDT1	IGD	0.0155	0.0169	0.1072
	Spacing	0.0193	0.0299	0.0382
	HV	0.7043	0.7032	0.5701
	Time(s)	3.4111	5.5565	9.5226
ZDT2	IGD	0.0016	0.0017	0.0484
	Spacing	0.002	0.0022	0.0356
	HV	0.4472	0.4471	0.3907
	Time(s)	4.0869	6.3615	9.1861
ZDT3	IGD	0.0019	0.0023	0.0169
	Spacing	0.002	0.0022	0.0307
	HV	0.6008	0.6006	0.5921
	Time(s)	4.6813	8.663	9.832
ZDT6	IGD	0.0013	0.0014	0.0095
	Spacing	0.0019	0.0031	0.0494
	HV	0.3906	0.3903	0.2332
	Time(s)	4.2113	5.9763	9.2985

4. Conclusions

This study presents a wind power interval-forecasting framework combining MBCT-SR feature fusion, KFCM clustering, TCN-Transformer forecasting, ABKDE, and IMODBO-based correction. Cluster-specific weights regulate predicted residuals to balance interval coverage and width. In the three reported turbine cases, TCN-Transformer has the lowest MAPE, while the RMSE ranking varies by day. The corrected intervals have lower PINAW than the reported ABKDE intervals in all six cases; coverage varies across cases, and some intervals remain below the nominal level. These results support the proposed category-dependent correction as a way to adjust the coverage-width trade-off. Future work will include systematic component ablation, matched evaluation against additional hybrid forecasters, and validation across longer periods and multiple sites. A time-stamped, fully specified implementation record would support reproduction of the operational forecasting protocol.

Author Contributions: Conceptualization, B.K. and G.D.; methodology, B.K.; software, B.K. and Y.X.; validation, B.K., Y.X., and X.Y.; formal analysis, B.K. and Z.W.; investigation, X.Y., Y.M., and Y.W.; resources, G.D.; data curation, B.K.; writing—original draft preparation, B.K.; writing—review and editing, G.D. and Y.X.; visualization, B.K.; supervision, G.D.; project administration, G.D.; funding acquisition, G.D. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Jiangxi Provincial Natural Science Foundation, grant number 20253BAC260014.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: During this revision, AI-assisted tools were used for language editing, translation, formatting, and consistency checking of author-provided text, data, code, and figure labels. The authors take full responsibility for the accuracy and integrity of the manuscript and responses.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pan, C.; Wen, S.; Zhu, M.; Ye, H.; Ma, J.; Jiang, S. Hedge Backpropagation Based Online LSTM Architecture for Ultra-Short-Term Wind Power Forecasting. *IEEE Trans. Power Syst.* **2024**, *39*, 4179–4192. [CrossRef]
2. Karaman, Ö.A. Prediction of Wind Power with Machine Learning Models. *Appl. Sci.* **2023**, *13*, 11455. [CrossRef]
3. Yu, F.; Dong, C.; Jiang, J. Summary of Research on Power Forecasting Technology of New Energy Generation. In Proceedings of the 2019 2nd Asia Conference on Energy and Environment Engineering (ACEEE), Hiroshima, Japan, 8–10 June 2019; pp. 49–53. [CrossRef]
4. Yang, B.; Zhu, T.; Cao, P.; Guo, Z.; Zeng, C.; Li, D.; Chen, Y.; Ye, H.; Shao, R.; Shu, H.; et al. Classification and Summarization of Solar Irradiance and Power Forecasting Methods: A Thorough Review. *CSEE J. Power Energy Syst.* **2023**, *9*, 978–995. [CrossRef]
5. Liu, X.; Yang, L.; Zhang, Z. Short-Term Multi-Step Ahead Wind Power Predictions Based on a Novel Deep Convolutional Recurrent Network Method. *IEEE Trans. Sustain. Energy* **2021**, *12*, 1820–1833. [CrossRef]
6. Kim, Y.; Hur, J. An Ensemble Forecasting Model of Wind Power Outputs Based on Improved Statistical Approaches. *Energies* **2020**, *13*, 1071. [CrossRef]
7. Abou Houran, M.; Salman Bukhari, S.M.; Zafar, M.H.; Mansoor, M.; Chen, W. COA-CNN-LSTM: Coati Optimization Algorithm-Based Hybrid Deep Learning Model for PV/Wind Power Forecasting in Smart Grid Applications. *Appl. Energy* **2023**, *349*, 121638. [CrossRef]
8. Ye, W.; Yang, D.; Tang, C.; Wang, W.; Liu, G. Combined Prediction of Wind Power in Extreme Weather Based on Time Series Adversarial Generation Networks. *IEEE Access* **2024**, *12*, 102660–102669. [CrossRef]
9. Deng, W.; Dai, Z.; Liu, X.; Chen, R.; Wang, H.; Zhou, B.; Tian, W.; Lu, S.; Zhang, X. Short-Term Wind Power Prediction Based on Wind Speed Interval Division and TimeGAN for Gale Weather. In Proceedings of the 2023 International Conference on Power Energy Systems and Applications (ICoPESA), Nanjing, China, 24–26 February 2023; pp. 352–357. [CrossRef]
10. Yu, G.Z.; Lu, L.; Tang, B.; Wang, S.Y.; Chung, C.Y. Ultra-Short-Term Wind Power Subsection Forecasting Method Based on Extreme Weather. *IEEE Trans. Power Syst.* **2023**, *38*, 5045–5056. [CrossRef]
11. Uykan, Z. Fusion of Centroid-Based Clustering with Graph Clustering: An Expectation-Maximization-Based Hybrid Clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 4068–4082. [CrossRef] [PubMed]
12. Bae, K.Y.; Jang, H.S.; Sung, D.K. Hourly Solar Irradiance Prediction Based on Support Vector Machine and Its Error Analysis. *IEEE Trans. Power Syst.* **2017**, *32*, 935–945. [CrossRef]
13. Chokr, B.; Chatti, N.; Charki, A.; Lemenand, T.; Hammoud, M. Feature Extraction-Reduction and Machine Learning for Fault Diagnosis in PV Panels. *Sol. Energy* **2023**, *262*, 111918. [CrossRef]
14. Chen, Z.; Zhang, B.; Du, C.; Meng, W.; Meng, A. A Novel Dynamic Spatio-Temporal Graph Convolutional Network for Wind Speed Interval Prediction. *Energy* **2024**, *294*, 130930. [CrossRef]
15. Hou, G.; Wang, J.; Fan, Y.; Zhang, J.; Huang, C. A Novel Wind Power Deterministic and Interval Prediction Framework Based on the Critic Weight Method, Improved Northern Goshawk Optimization, and Kernel Density Estimation. *Renew. Energy* **2024**, *226*, 120360. [CrossRef]
16. Niu, D.; Sun, L.; Yu, M.; Wang, K. Point and Interval Forecasting of Ultra-Short-Term Wind Power Based on a Data-Driven Method and Hybrid Deep Learning Model. *Energy* **2022**, *254*, 124384. [CrossRef]
17. Ma, C.; Xu, X.; Pang, X.; Li, X.; Zhang, P.; Liu, L. Scenario-Based Ultra-Short-Term Rolling Optimal Operation of a Photovoltaic-Energy Storage System under Forecast Uncertainty. *Appl. Energy* **2024**, *356*, 122425. [CrossRef]
18. Massidda, L.; Bettio, F.; Marrocu, M. Probabilistic Day-Ahead Prediction of PV Generation. A Comparative Analysis of Forecasting Methodologies and of the Factors Influencing Accuracy. *Sol. Energy* **2024**, *271*, 112422. [CrossRef]
19. Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A.F. Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals. *IEEE Trans. Neural Netw.* **2011**, *22*, 337–346. [CrossRef] [PubMed]
20. Pinson, P.; Tastu, J. Discussion of "Prediction Intervals for Short-Term Wind Farm Generation Forecasts" and "Combined Nonparametric Prediction Intervals for Wind Power Generation". *IEEE Trans. Sustain. Energy* **2014**, *5*, 1019–1020. [CrossRef]
21. Yin, H.; Yang, M.; Lan, H.; Hong, Y.-Y.; Guo, D.; Jin, F. A Hybrid Graph Attention Network Based Method for Interval Prediction of Shipboard Solar Irradiation. *Energy* **2024**, *298*, 131131. [CrossRef]
22. Wen, Y.; AlHakeem, D.; Mandal, P.; Chakraborty, S.; Wu, Y.-K.; Senjyu, T.; Paudyal, S.; Tseng, T.-L. Performance Evaluation of Probabilistic Methods Based on Bootstrap and Quantile Regression to Quantify PV Power Point Forecast Uncertainty. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 1134–1144. [CrossRef] [PubMed]
23. Zhu, J.; He, Y.; Gao, Z. Wind Power Interval and Point Prediction Model Using Neural Network Based Multi-Objective Optimization. *Energy* **2023**, *283*, 129079. [CrossRef]
24. Su, H.-Y.; Lai, C.-C. Toward Improved Load Forecasting in Smart Grids: A Robust Deep Ensemble Learning Framework. *IEEE Trans. Smart Grid* **2024**, *15*, 4292–4296. [CrossRef]
25. Li, Z.; Ye, L.; Zhao, Y.; Song, X.; Teng, J.; Jin, J. Short-Term Wind Power Prediction Based on Extreme Learning Machine with Error Correction. *Prot. Control Mod. Power Syst.* **2016**, *1*, 1. [CrossRef]

26. Su, H.-Y.; Tang, C. Dynamic-Error-Compensation-Assisted Deep Learning Framework for Solar Power Forecasting. *IEEE Trans. Sustain. Energy* **2022**, *13*, 1865–1868. [CrossRef]
27. Phan, Q.-T.; Wu, Y.-K.; Phan, Q.-D.; Lo, H.-Y. A Novel Forecasting Model for Solar Power Generation by a Deep Learning Framework with Data Preprocessing and Postprocessing. *IEEE Trans. Ind. Appl.* **2023**, *59*, 220–231. [CrossRef]
28. Li, M.; Yang, M.; Yu, Y.; Lee, W.-J. A Wind Speed Correction Method Based on Modified Hidden Markov Model for Enhancing Wind Power Forecast. *IEEE Trans. Ind. Appl.* **2022**, *58*, 656–666. [CrossRef]
29. Xie, T.; Yang, L.; Zhang, G.; Zhang, K. Research on a Wind Power Interval Prediction Method Coupling Process Characteristics and Deep Transfer Learning. *Energy Convers. Manag. X* **2025**, *28*, 101342. [CrossRef]
30. Quan, H.; Meng, Y. Wind Power Interval Forecasting via a Transfer Learning-Based Two-Stage Method with Multi-Objective Optimization. *Appl. Soft Comput.* **2026**, *201*, 115616. [CrossRef]
31. Fu, X.-Y.; Feng, Z.-K.; Niu, W.-J. A Twin Model Framework with Physical Features and Multi-Objective Optimization for Point and Interval Predictions of Wind Power. *Energy* **2026**, *358*, 141409. [CrossRef]
32. Li, D.; Liu, C. Study on the Universality of the Normal Cloud Model. *Strateg. Study CAE* **2004**, *6*, 28–34.
33. Xue, J.; Shen, B. Dung Beetle Optimizer: A New Meta-Heuristic Algorithm for Global Optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.