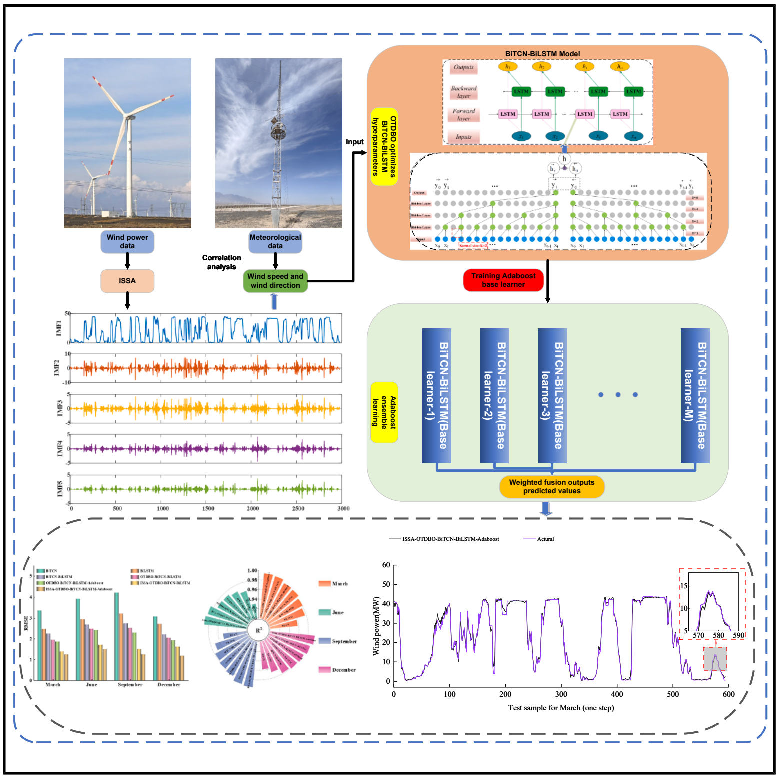
iScience

Article

Adaptive singular spectral decomposition hybrid framework with quadratic error correction for wind power prediction

Graphical abstract


BiTCN-BiLSTM Model
Outputs
Backward layer
Forward layer
Inputs
LSTM
OTDBO optimizes BiTCN-BiLSTM hyperparameters
Input
Wind power data
Meteorological data
Correlation analysis
ISSA
Wind speed and wind direction
Training Adaboost base learner
Adaboost ensemble learning
BiTCN-BiLSTM(Base learner-1)
BiTCN-BiLSTM(Base learner-2)
BiTCN-BiLSTM(Base learner-3)
BiTCN-BiLSTM(Base learner-M)
Weighted fusion outputs predicted values
IMF1
IMF2
IMF3
IMF4
IMF5
RMSE
March
June
September
December
R²
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost
Actual
Wind power(MW)
Test sample for March (one step)

Authors

Chunliang Mai, Lixin Zhang, Omar Behar, Xue Hu, Xuewei Chao

Correspondence

zhlx2001329@163.com (L.Z.),
huxue@foxmail.com (X.H.)

In brief

Applied computing; Applied sciences; Energy engineering

Highlights

- ISSA-OTDBO-BiTCN-BiLSTM-Adaboost improves wind power forecasting accuracy
- Enhances OTDBO's population diversity and global search for better optimization
- Uses ISSA to decompose wind power data, precisely extracting key features
- Combines BiTCN-BiLSTM and Adaboost, achieving strong multi-level forecasting gains

Mai et al., 2025, iScience *28*, 112360
May 16, 2025 © 2025 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2025.112360

CellPress

iScience

CellPress
OPEN ACCESS

Article

Adaptive singular spectral decomposition hybrid framework with quadratic error correction for wind power prediction

Chunliang Mai,[1,2] Lixin Zhang,[3,4,6,*] Omar Behar,[5] Xue Hu,[1,4,*] and Xuewei Chao[3,4]
[1]College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832003, China
[2]Key Laboratory of Northwest Agricultural Equipment, Ministry of Agriculture and Rural Affairs, Shihezi 832003, China
[3]Bingtuan Energy Development Institute, Shihezi University, Shihezi 832000, China
[4]Xinjiang Production & Construction Corps Key Laboratory of Advanced Energy Storage Materials and Technologies, Shihezi University, Shihezi 832000, China
[5]King Abdullah University of Science and Technology, Thuwal, Jeddah 23955-6900, Saudi Arabia
[6]Lead contact
*Correspondence: zhlx2001329@163.com (L.Z.), huxue@foxmail.com (X.H.)
https://doi.org/10.1016/j.isci.2025.112360

SUMMARY

High-precision wind power forecasting is essential for grid scheduling and renewable energy utilization. Wind data's nonlinear, stochastic, and multi-scale characteristics create prediction challenges. This study proposes a hybrid model integrating adaptive improved singular spectrum analysis (ISSA), optimized bidirectional temporal convolutional network–bidirectional long short-term memory (BiTCN-BiLSTM) networks, and AdaBoost ensemble learning. Adaptive ISSA provides parameter-free, data-driven modal decomposition to reduce noise. Hybrid strategy-enhanced dung beetle optimization (OTDBO) fine-tunes hyperparameters of BiTCN-BiLSTM, and AdaBoost dynamically corrects errors, significantly improving robustness. Tests using seasonal datasets from Dabancheng wind farm (China) show substantial performance improvement (mean absolute error [MAE] reduced by 45.4%, root-mean-square error (RMSE) by 47.6%, $p < 0.001$), and training time reduced by 12.1%–21.3%. This method offers accurate, scalable forecasting for reliable renewable energy integration.

INTRODUCTION

Background and motivation

As climate change intensifies, renewable energy adoption has become a key global strategy. Among various sources, wind power stands out for its abundance, environmental benefits, and economic viability.[1,2] However, the inherent randomness of wind speed and meteorological conditions introduces intermittency and volatility, challenging grid stability, and economic dispatch.[3] Accurate forecasting enhances real-time grid management, reduces failure risks, and optimizes energy markets, improving resource allocation, and trading efficiency. Additionally, precise predictions aid in reserve capacity planning and energy storage management, boosting overall economic and environmental sustainability.

Despite these benefits, the multimodal, non-stationary, and high-frequency nature of wind power data degrades prediction accuracy and model stability. Furthermore, challenges in model construction and hyperparameter tuning, including local optima, often hinder forecasting performance, posing a significant bottleneck to high-precision wind power prediction.[4,5]

Literature review

Various algorithms and deep learning models have been developed for wind power forecasting. Broadly, existing approaches fall into physical, statistical, machine learning, and hybrid models.[6] Physical models leverage numerical weather prediction parameters (e.g., atmospheric pressure, temperature, and humidity) to estimate power output. Even with limited historical data, they provide reliable forecasts for new or planned wind farms. Additionally, their focus on geographical and topographical factors makes them effective for long-term predictions in complex terrains.[7] However, these models are computationally intensive, requiring long processing times and high computational resources. Their intricate modeling process also limits adaptability to rapidly changing conditions.[8]

Statistical models establish relationships between independent and dependent variables using historical wind power data.[9,10] Common methods, include AutoRegressive Moving Average (ARMA),[11] AutoRegressive Integrated Moving Average (ARIMA),[12] and Seasonal ARIMA (SARIMA),[13] which offer simplicity, computational efficiency, and strong interpretability. For example, ref.[14] introduced a generalized AutoRegressive Moving Average with eXogenous inputs model (ARMAX) for a 2.1 kW grid-connected PV system, significantly outperforming ARIMA. Similarly, ref.[15] proposed an online SARIMA variant for hourly load forecasting in northern Vietnam, achieving a 4.57% average absolute percentage error.

However, traditional statistical models rely on stable autocorrelations and explicit input-output relationships, limiting adaptability

Table 1. Comparative analysis of existing literature on wind energy prediction

Reference	Module	Horizon	Dataset	Challenges
Singh et al.[12]	RWT-ARIMA	1, 3, 5, 7, 10 min	Ireland	1. Limited prediction strategy; 2. Inadequate handling of data complexity and nonlinearity.
Yang et al.[23]	SVM-enhanced Markov model	10 min	Xcel Energy	1. Limited prediction performance; 2. Lack of generalization and robustness.
Qu et al.[30]	CEEMDAN-EWT-FPA-BP	15, 30, 45 min	French wind farm	Limited dataset time span.
Shams et al.[33]	Multi-task temporal feature attention-based LSTM (MTTFA-LSTM)	1 h	national renewable energy laboratory	Prediction performance heavily depends on loss weights.
Fantini et al.[34]	Stationary Wavelet Transform with GRU (SWT-GRU)	1 h	Brazil	Lack of comparative experiments on decomposition methods.
Karijadi et al.[44]	CEEMDAN-EWT-LSTM	10 min	France, Turkey	Secondary decomposition increases the workload of prediction tasks.
Li et al.[57]	BWO-BiLSTM-ATT	1 h	Guangdong, China	Lack of multi-step prediction experiments.
Wang et al.[52]	CEEMDAN-SE-TR-BiGRU-Attention	15 min	Ningxia, China	Insufficient consideration of additional influencing factors.
The proposed method	ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	15, 30, 45 min	Xinjiang, China	Further research is required.

This table compares various wind energy prediction methods from different studies, focusing on their forecasting horizon, datasets, and challenges.

to changing conditions. To address this, improved methods have emerged. Ref.[16] developed an enhanced ARIMA to better capture temporal dependencies and probability distributions, while ref.[17] integrated ARIMA with random forest (RF) and Boosted Classification and Regression Tree (BCART) for multi-horizon wind power forecasting, significantly boosting accuracy.

While statistical models excel in short-term forecasting, they struggle with non-stationarity and randomness in wind power data. Integrating them with other methods enhances robustness and accuracy. However, these models rely heavily on historical data, failing to capture nonlinear and complex relationships. They also lack real-time adaptability, are sensitive to outliers, and ignore spatiotemporal correlations. To address these limitations, research increasingly explores machine learning and hybrid models to improve forecasting performance.[18,19]

With advancements in computer science, machine learning models have been widely adopted for forecasting due to their ability to capture nonlinear relationships and their self-learning capabilities.[20,21] Compared to physical and statistical models, they excel at extracting key features from multidimensional data.[22] Representative models include Support Vector Machine (SVM),[23] Least Squares Support Vector Machine (LSSVM),[24] Extreme Learning Machine (ELM),[25] Back Propagation Neural Network (BPNN),[26] Wavelet Neural Network (WNN),[27] and General Regression Neural Network (GRNN).[28] For instance, ref.[29] introduced an LSSVM-GSA model, where GSA-optimized LSSVM outperformed BPNN and SVM in short-term forecasting. Similarly, ref.[30] employed Enhanced Crow Search Algorithm (ENCSA) to optimize ELM, achieving superior accuracy across multiple error metrics. Additionally, ref.[31] proposed a hybrid neural network combining Long Short-Term Memory (LSTM), SVM, BPNN, and ELM for wind and solar forecasting, demonstrating better performance in mean absolute percentage error (MAPE), mean absolute error (MAE), mean squared error (MSE), and RMSE.

Deep learning has emerged as a mainstream approach in wind power forecasting, effectively modeling complex nonlinear time series and achieving superior performance. Key models include Recurrent Neural Network (RNN),[32] LSTM,[33] Gated Recurrent Unit (GRU),[34] Convolutional Neural Network (CNN),[35] and the attention mechanism.[36] Unlike traditional machine learning, which requires extensive feature engineering, deep learning primarily involves data preprocessing, network design, and hyperparameter tuning, enabling end-to-end forecasting with enhanced nonlinear pattern capture. However, single deep learning models may experience accuracy loss due to input fluctuations.[37] To address this, research has increasingly focused on hybrid models, integrating data preprocessing, parameter optimization, and model fusion to overcome individual model limitations. As a result, hybrid approaches are gradually surpassing standalone models, becoming a key direction in wind power forecasting.[38]

Integrating data decomposition with deep learning and intelligent optimization has become a key focus in hybrid model research. Data decomposition simplifies complex wind power datasets, revealing hidden patterns, trends, and periodicities, thereby enhancing forecasting accuracy and reliability.[39] Common methods include Empirical Mode Decomposition (EMD),[40] Ensemble Empirical Mode Decomposition (EEMD),[41] Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN),[42] Variational Mode Decomposition (VMD),[43] and singular spectrum analysis. Although EMD requires no predefined basis functions, it suffers from mode mixing. EEMD mitigates this by adding white noise, but tuning noise amplitude and ensemble averaging parameters is empirical. VMD also requires careful regularization and mode selection, lacking a standardized approach, which affects the stability and adaptability of decomposition results.

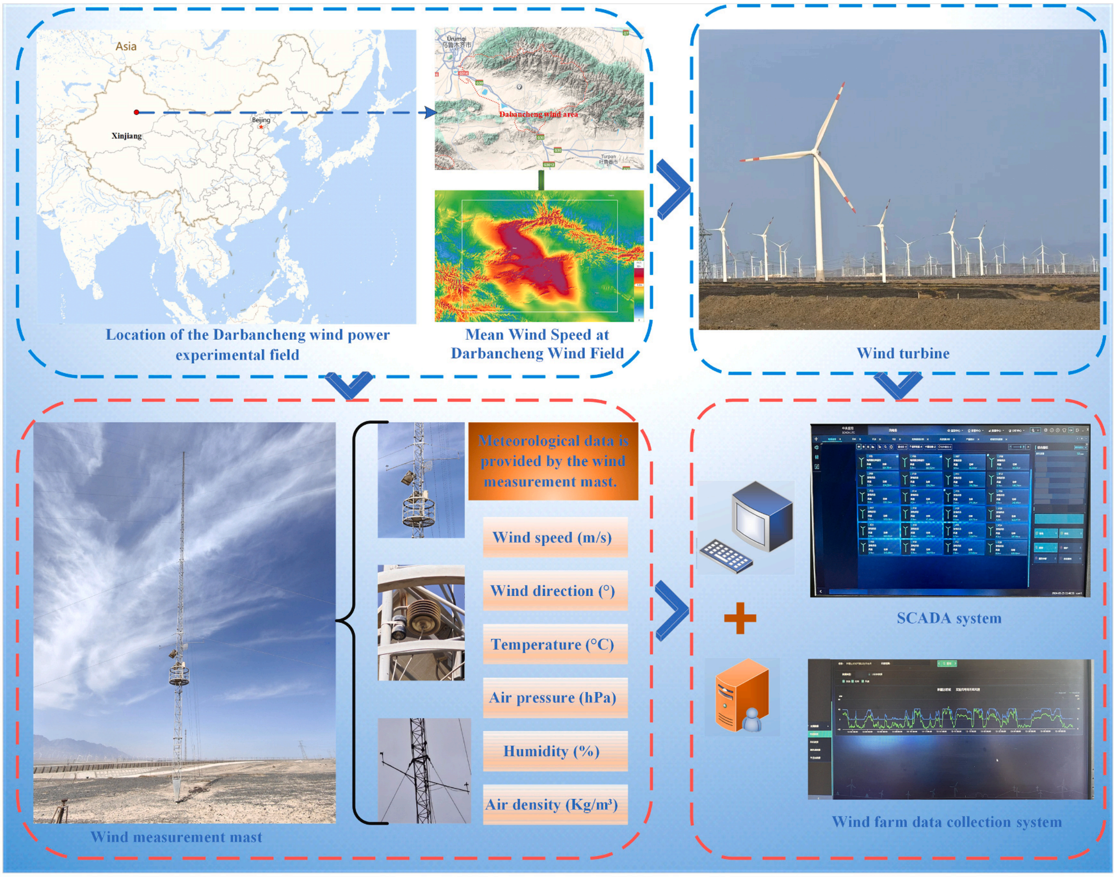


Figure 1. Spatial distribution and data acquisition system of the Dabancheng wind farm

The system includes wind turbines, a measurement mast for meteorological data collection (e.g., wind speed, temperature, humidity), and an SCADA system for data monitoring and storage.

To tackle these challenges, researchers have explored secondary decomposition and optimization strategies. Ref.[44] combined CEEMDAN with empirical wavelet transform for preprocessing, while ref.[45] integrated CEEMDAN and VMD, applying kernel principal component analysis for dimensionality reduction. Ref. [46] further optimized VMD parameters using the flower pollination algorithm. These approaches have enhanced forecasting accuracy and interpretability, but further improvements are needed to adapt models to diverse data distributions and complex wind farm scenarios.

Deep learning has advanced wind power forecasting, but its performance heavily depends on hyperparameter optimization. Proper selection enhances accuracy and generalization, yet the high-dimensional, non-convex, and nonlinear nature of hyperparameter space makes optimization challenging.[47] Traditional methods like grid search and random search offer simple tuning strategies. Grid search systematically evaluates all combinations but becomes computationally impractical for complex models.[48] Random search is more efficient in high-dimensional spaces[49] but may fail to converge optimally within a limited computational budget. To overcome these limitations, Bayesian optimization (BO) has gained popularity due to its use of surrogate models (e.g., Gaussian processes) and intelligent sampling.[50] However, BO's sequential nature hinders parallelization, and as dimensionality or non-stationarity increases, its surrogate model struggles to approximate the objective function, reducing search accuracy and stability.[51]

To address this bottleneck, researchers have introduced swarm intelligence algorithms, such as particle swarm optimization (PSO)[52] and genetic algorithm (GA).[53] These population-based strategies leverage distributed exploration to avoid local optima and support parallel computing. For instance, ref[54] used PSO to optimize SVM kernel parameters, improving classification accuracy, while[55] applied GA to optimize BPNN weights, enhancing wind speed prediction. Other intelligent optimization algorithms, including Greylag Goose Optimization,[56] Whale Optimization Algorithm (WOA),[57] Sparrow Search Algorithm,[58] and Dung Beetle Optimization (DBO),[59] have also been widely used for parameter tuning. These methods vary in search speed and convergence accuracy, yet slow convergence and high parameter sensitivity remain challenges, limiting their effectiveness in wind power forecasting. Thus, further algorithmic improvements are needed for efficiency and robustness.

Moreover, hybrid models have been developed to enhance forecasting accuracy. Ref.[60] proposed a CapSA-VMD-ResNet-GRU-Attention model, while ref.[61] introduced the VMD-ADE-TFT

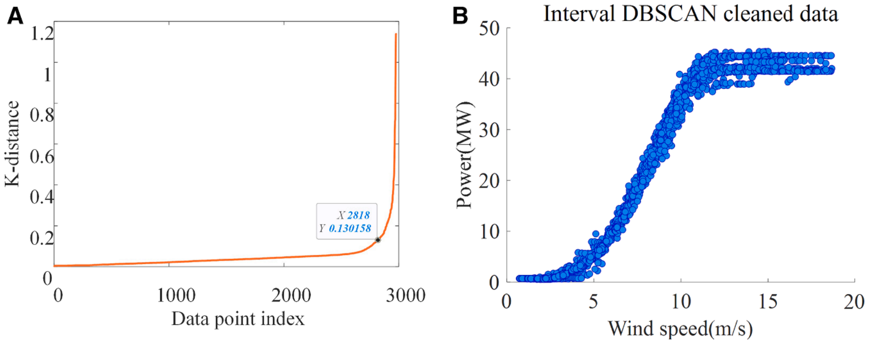


Figure 2. Interval DBSCAN data cleaning

(A) k-distance curve used to determine the optimal clustering threshold.

(B) Wind speed-power distribution after outlier removal with Interval DBSCAN.

model, achieving superior forecasting across eight case studies. Similarly, ref.[62] developed CEEMDAN-SE-TR-BiGRU-Attention, integrating multiple techniques to significantly improve predictive performance.

While hybrid wind power forecasting has achieved success in various applications, significant deficiencies remain in model development. Many methods merely combine models or optimize parameters, overlooking the synergy between data decomposition and backend optimization. A single-dimensional optimization approach often fails to handle the complexities and variabilities of wind power data, particularly in multi-step forecasting, where performance may degrade or lead to model failure. Moreover, despite promising results from hybrid deep learning models like CNN-LSTM and Transformer-BiGRU, these approaches primarily rely on serial or parallel deep learning combinations, lacking integration with ensemble

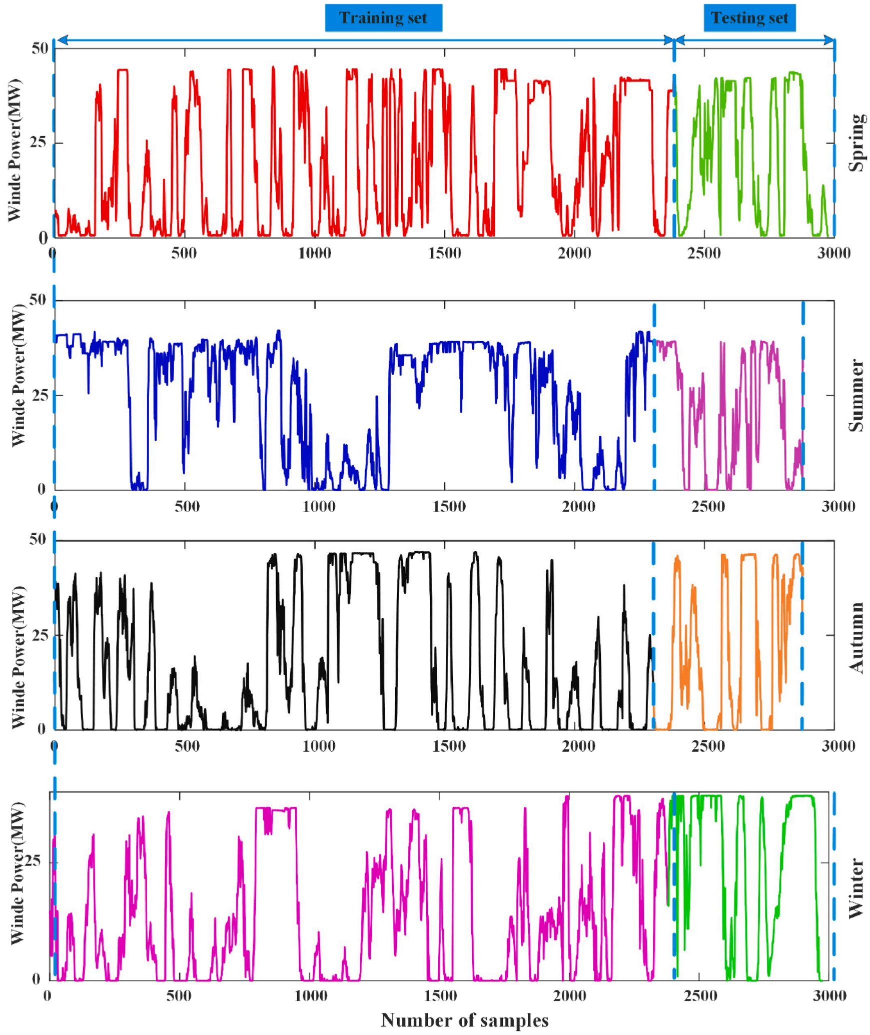


Figure 3. Distribution of historical wind power data across different seasons in the Dabancheng area

Wind power time series data are segmented into training and testing sets for Spring, Summer, Autumn, and Winter.

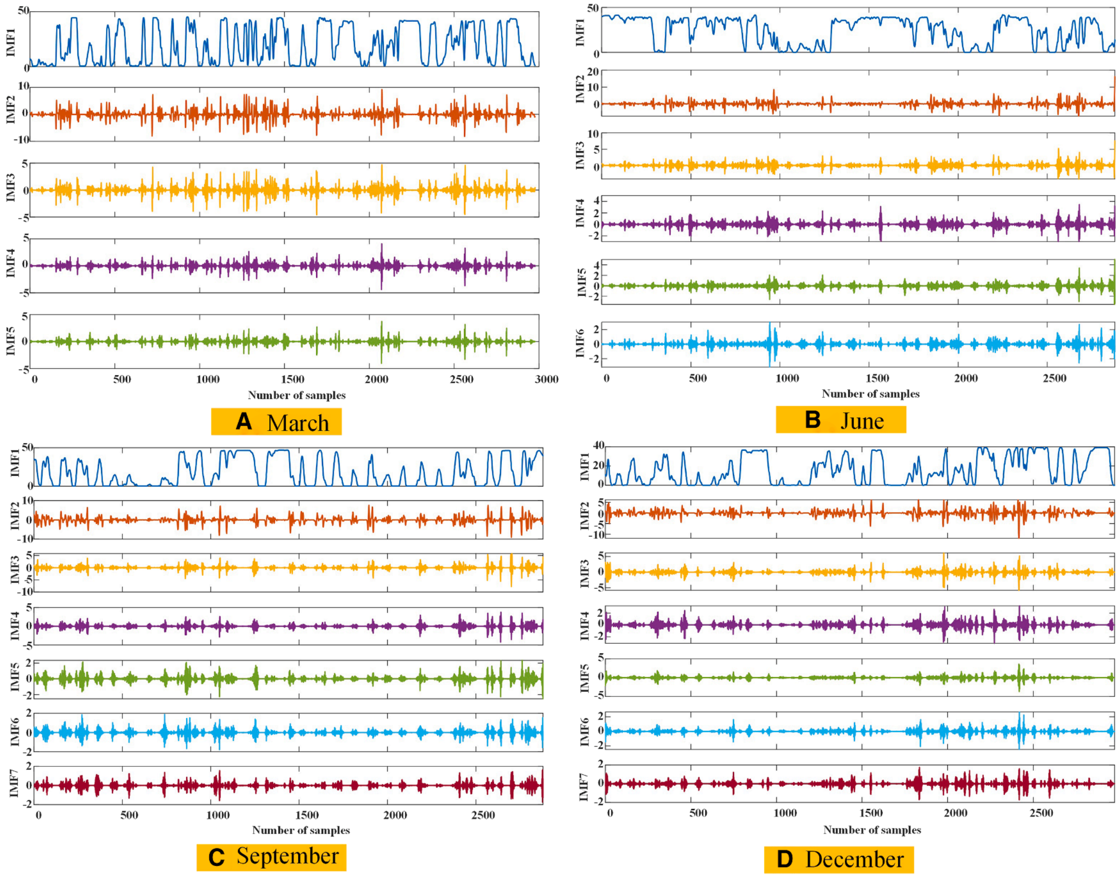


Figure 4. ISSA mode decomposition results across four seasons

ISSA decomposes wind power data into intrinsic mode functions (IMFs) for different seasons: (A) March, (B) June, (C) September, and (D) December.

learning. Dynamic ensemble mechanisms, which iteratively adjust sample weights and multi-model fusion, can fully leverage model strengths, mitigating the limitations of single-model optimization. Table 1 provides a comparative analysis of existing wind energy forecasting methods.

This study proposes a deep hybrid forecasting model—ISSA-OTDBO-BiTCN-BiLSTM-Adaboost—integrating dual optimization and ensemble learning to enhance wind power forecasting accuracy and stability. The model employs adaptive singular spectrum analysis (ISSA) for data-driven decomposition, eliminating reliance on manual parameter tuning. Optimized dung beetle algorithm (OTDBO) ensures efficient hyperparameter tuning, improving pattern recognition, while Adaboost ensemble learning dynamically adjusts sample weights to minimize accumulated errors. Experiments on seasonal datasets confirm the model's superiority over existing methods, highlighting its adaptability and practical value. This provides a reliable, high-performance solution for forecasting renewable energy generation.

Contributions and paper organization

The main contributions of this work are as follows.

1. Proposed ISSA-OTDBO-BiTCN-BiLSTM-Adaboost, a hybrid forecasting model integrating improved (ISSA) for data decomposition and OTDBO for hyperparameter tuning, achieving fine-grained decomposition, deep feature extraction, and multi-level forecasting, significantly improving multi-step wind power prediction accuracy and robustness.
2. Enhanced DBO by incorporating Kent mapping-based initialization, Osprey algorithm-inspired global exploration, and adaptive t-distribution perturbations, effectively avoiding local optima, and boosting search efficiency.
3. Developed an autocorrelation-guided ISSA technique to automatically determine decomposition modes, balancing local details and long-term trends, reducing reliance on manually set parameters.
4. Integrated BiTCN and BiLSTM with Adaboost, incorporating dynamic error correction to strengthen multi-level predictive capabilities for complex, non-stationary time series.
5. Conducted extensive comparative and ablation experiments across multiple seasons, demonstrating superior forecasting accuracy, stability, and generalization over traditional hybrid models.

The remainder of this paper is structured as follows: section 2 introduces the fundamental theories and proposed forecasting

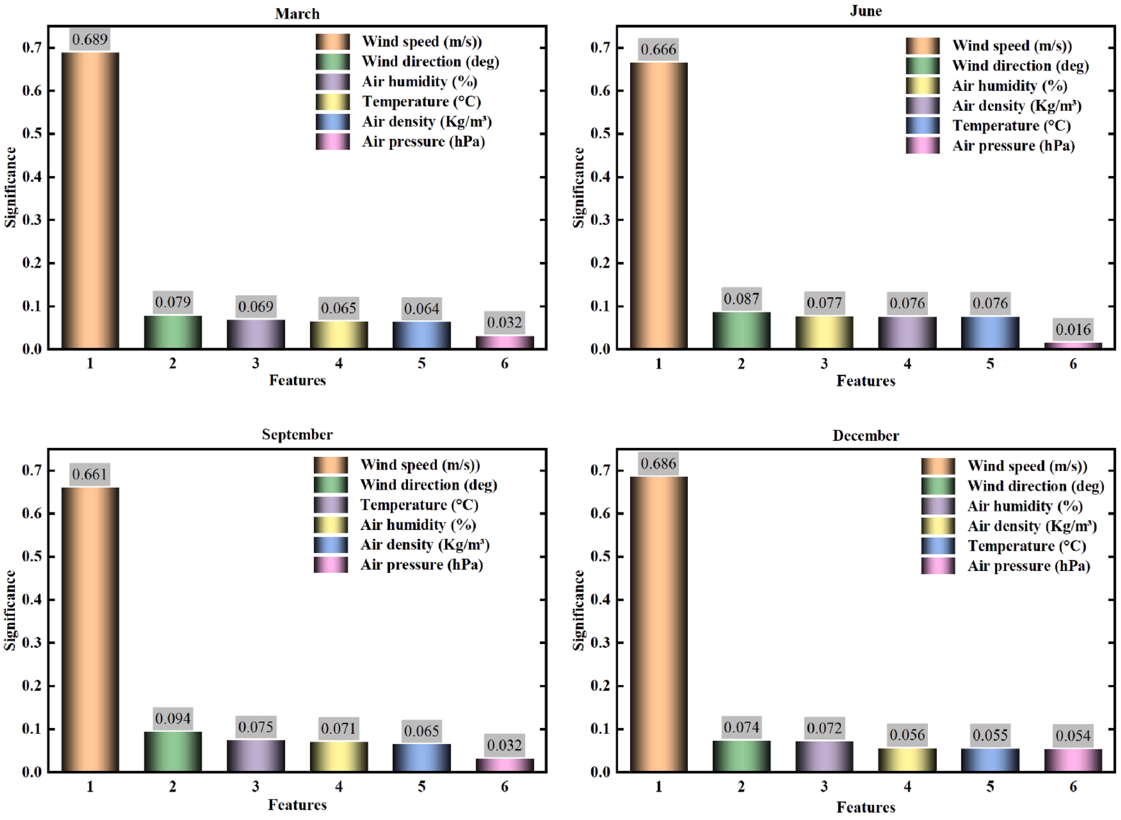


Figure 5. RF feature importance ranking in different seasons

The significance of meteorological variables in wind power prediction is ranked for different seasons using random forest.

model, section 3 details the implementation of ISSA-OTDBO-BiTCN-BiLSTM-Adaboost, section 4 covers data preprocessing, model configuration, and evaluation metrics, section 5 presents and analyzes comparative experimental results, and section 6 summarizes key findings and discusses future research directions.

RESULTS AND DISCUSSION

Data source and preprocessing

The proposed model is evaluated using 2022 data from the Dabancheng Experimental Wind Farm in Xinjiang, situated between the eastern Tianshan Mountains and the southern Bogda foothills, a region rich in wind energy (Figure 1 illustrates spatial distribution and data collection). The SCADA system records 15-min intervals, generating 96 daily power records. The dataset includes meteorological features such as wind speed, wind direction, air pressure, temperature, humidity, air density, and actual power, collected via wind measurement tower sensors.

Due to seasonal variations, data are categorized into: spring (Feb-Apr 2022), summer (May-Jul 2022), autumn (Aug-Oct 2022), and winter (Nov 2022-Jan 2023). To assess generalization capability, data from different seasons is split 80% for training and 20% for testing.

The model training and forecasting experiments were conducted on Windows 11 using an Intel Core i9-12900H (2.5GHz) processor and an NVIDIA GeForce RTX 3060 GPU, ensuring ample computational and memory resources. The software environment is MATLAB 2023b, with GPU parallel acceleration via the Parallel Computing Toolbox, significantly enhancing large-scale data processing and deep model training efficiency. This hardware setup maximizes feature extraction and parallel computing, improving training and inference speed. Additionally, its high scalability supports future deployment in larger or more diverse datasets.

Anomalous data in the wind turbine acquisition system, caused by equipment failures or extreme weather, can impact forecasting accuracy. To improve data quality, this study integrates DBSCAN clustering and interval-based segmentation for anomaly detection and correction. Figure 2A illustrates the k-distance curve, where the elbow point determines epsilon, with minPts set to 12. To enhance detection accuracy, wind speed is segmented into 0.5 m/s intervals, and DBSCAN is applied separately to identify normal vs. anomalous data. Anomalies are corrected via linear interpolation, ensuring data continuity. Figure 2B presents the cleaned k-distance curve and wind speed-power distribution, while Figure 3 highlights seasonal variations in Dabancheng wind power, revealing significant fluctuations across seasons.

Parameter settings

This study introduces an autocorrelation function threshold (θ) in ISSA to regulate decomposition precision and noise suppression. Experimental comparisons show that at θ = 0.8, the

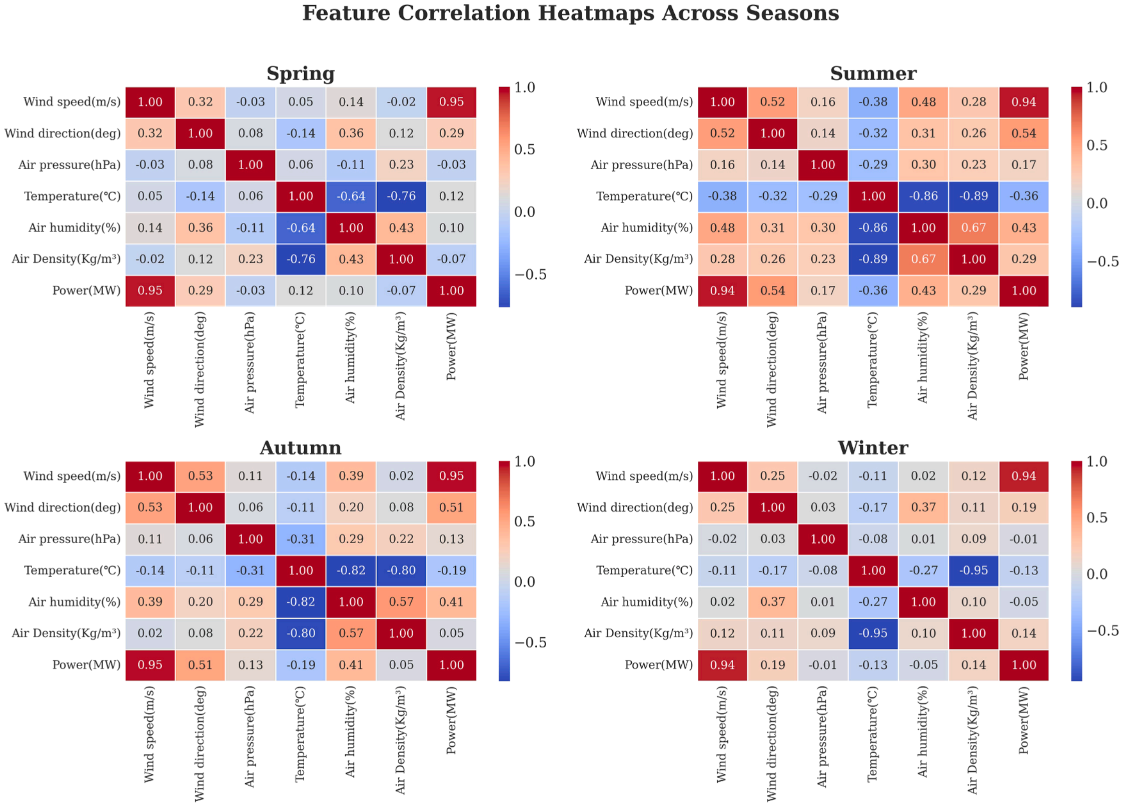


Figure 6. Seasonal correlation heatmaps of meteorological variables

Correlation matrices show relationships between wind power and meteorological factors (e.g., wind speed, temperature) across four seasons.

decomposed sub-sequences more effectively capture key wind power features while minimizing reconstruction error. Thus, θ = 0.8 is selected as the optimal threshold. Figure 4 presents the decomposition results across four seasonal datasets.

Wind power forecasting requires effective feature selection to reduce computational costs and improve prediction accuracy. This study applies RF and correlation heatmap analysis to optimize input variables. RF, using 100 decision trees, ranks feature importance via out-of-bag (OOB) error (Figure 5). Results indicate wind speed as the most influential factor (importance >0.60) with a strong correlation to wind power, followed by wind direction. Other meteorological variables (air pressure, temperature, humidity, and air density) show seasonal variability.

The correlation heatmap (Figure 6) further confirms wind speed's consistently strong correlation with wind power (>0.80) across all seasons, while wind direction's importance varies, requiring adaptive modeling. Thus, wind speed and wind direction are selected as model inputs to enhance efficiency and predictive accuracy.

To assess the convergence and generalization of the proposed model, training and testing loss curves for different seasons are plotted in Figure 7. Both curves show a rapid decline in the first 50 epochs and stabilize around epoch 100, confirming effective learning and convergence. The minimal gap between training and testing losses indicates strong generalization without significant overfitting. Additionally, the consistently low

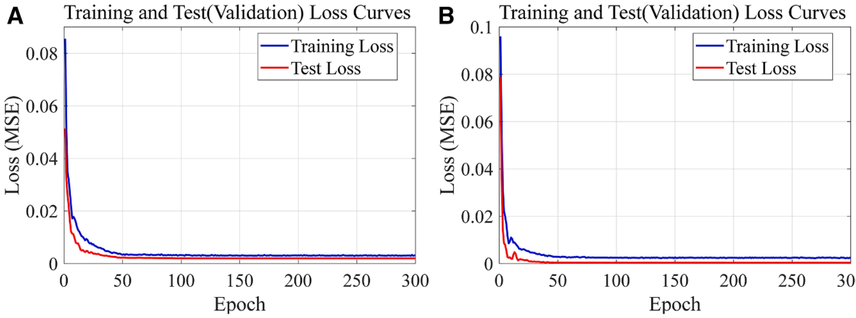


Figure 7. Training and testing loss curves of the proposed model on Spring and Winter datasets

(A) Spring dataset loss curves.

(B) Winter dataset loss curves, showing model convergence during training and testing.

Table 2. Impact of different hyperparameters on prediction accuracy

Experiment	Learning rate	Number of hidden neurons	Number of filters	L2Regularization	Batch Size	Mean absolute error (MAE)
Test 1	0.01	25	20	0.001	32	1.65
Test 2	0.005	25	20	0.001	32	1.94
Test 3	0.01	25	20	0.001	64	1.69
Test 4	0.01	50	80	0.001	32	2.58
Test 5	0.01	100	50	0.001	32	2.43
GA Optimization	0.0095	30	32	0.0015	32	1.45
Bayesian Optimization	0.0068	28	29	0.0012	32	1.55
OTDBO Optimization	0.0081	35	27	0.0013	32	1.13

The table presents the impact of various hyperparameter configurations on model performance, measured by mean absolute error (MAE).

final loss values across seasons highlight the model's robustness in handling seasonal variations in wind power forecasting.

To evaluate hyperparameter effects on prediction accuracy, preliminary experiments on the March dataset tested different BiTCN-BiLSTM configurations (Table 2). Results showed that excessive neurons led to overfitting, while an inappropriate batch size reduced accuracy. For instance, increasing BiLSTM neurons to 50 or 100 (tests 4 and 5) raised MAE, indicating overfitting. A smaller batch size (32) outperformed a larger one (64, test 3), suggesting better generalization with smaller batches. Learning rate also significantly impacted convergence—a lower rate (0.005, test 2) slowed convergence and increased errors, whereas the OTDBO-optimized rate (0.0081) balanced stability and error control, achieving a lowest MAE of 1.13, outperforming manual tuning and conventional methods.

Table 3. Parameter settings for OTDBO-BiTCN-BiLSTM-Adaboost model

Models	Parameter	Value
BiTCN-BiLSTM	Learning rate	0.01
	Regularization	0.001
	Filter size	5
	Number of filters	32
	Dropout Factor	0.006
	Number of neurons in the hidden layer	50
OTDBO-BiTCN-BiLSTM-Adaboost	Learning rate	(0.001–0.3)
	Regularization	(0.0001–0.005)
	Population size	30
	Number of iterations	10
	Filter size	5
	Number of filters	(20–120)
	Dropout Factor	0.006
	Number of neurons in the hidden layer	(10–150)
	The number of mgm.throttletms. com/throttlemmweak predictors	5

This table lists the key hyperparameters used in the proposed OTDBO-BiTCN-BiLSTM-Adaboost model, including learning rate, population size, and number of iterations.

The OTDBO-based hyperparameter optimization for BiTCN-BiLSTM employs root-mean-square error (RMSE) as the fitness function. Optimization is performed within a predefined search space, including a learning rate (0.001, 0.3), BiLSTM hidden neurons (10, 150), filters (20, 120), and a regularization parameter (0.0001, 0.005). In each iteration, OTDBO adjusts hyperparameters and evaluates RMSE on the validation set, continuing until the maximum iterations are reached. Table 3 presents the detailed parameter settings for OTDBO-BiTCN-BiLSTM.

Concurrently, this study employs a rolling prediction strategy whereby the power output for subsequent time steps (1 step, 2 steps, 3 steps) is forecasted using historical meteorological and power data from the preceding 2 h (8 sampling points).

To assess the OTDBO-BiTCN-BiLSTM model for short-term wind power forecasting, experiments are conducted in five key areas: (1) comparison with benchmark models, (2) evaluation of modal decomposition methods, (3) assessment of optimization algorithms, (4) analysis of Adaboost's error correction effect, and (5) comparison with existing studies. Additionally, a multi-step forecasting task is designed to examine the model's ability to capture lagged dependencies and dynamic variations in the time series.

Evaluation of the prediction results in comparison with baseline model

Figure 8 presents the one-step forecasting results for the March dataset. The predicted values from ISSA-OTDBO-BiTCN-BiLSTM-Adaboost closely align with actual values, outperforming 10 benchmark models. Notably, in both power fluctuation and stable intervals, the proposed model accurately tracks power variations, whereas others exhibit significant deviations, either failing to capture sudden changes or incorrectly estimating fluctuation directions.

Additionally, the zoomed-in region in Figure 8 provides further insights, showing that the proposed model responds rapidly to abrupt power variations, demonstrating superior ability to capture complex nonlinear temporal features.

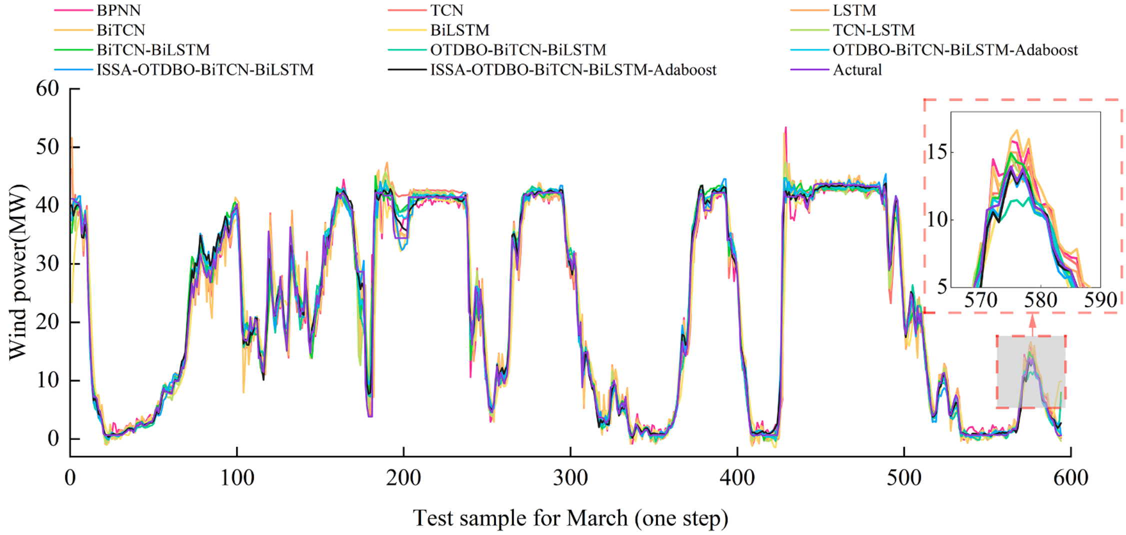


Figure 8. One-step-ahead prediction results for the March dataset

Wind power predictions of the proposed model and baseline models are compared with actual values. The inset highlights a zoomed-in section.

Table 4 shows that as models undergo optimization, prediction errors gradually decrease, significantly enhancing performance. Compared to TCN-LSTM, BiTCN-BiLSTM achieves a 12.22% MAE reduction, a 3.03% RMSE decrease, and a 0.18% R^2 increase for 1–3 step forecasting, demonstrating that the bidirectional structure and dilated convolution improve long-term

Table 4. Multi-step forecasting performance comparison (March dataset)

	March								
	MAE/IR_{MAE}			$RMSE/IR_{RMSE}$			$R^2/IR_R{}^2$		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BPNN	2.17	3.18	4.43	3.62	5.04	6.62	95.11%	90.48%	83.63%
	(64.9%)	(71.0%)	(71.2%)	(65.6%)	(71.3%)	(71.1%)	(4.5%)	(9.7%)	(15.2%)
TCN	1.92	3.02	4.19	3.32	5.12	6.57	95.88%	90.19%	83.90%
	(60.2%)	(69.4%)	(69.5%)	(62.5%)	(71.7%)	(70.9%)	(3.7%)	(10.0%)	(14.7%)
LSTM	1.87	3.27	4.47	3.18	5.43	6.88	96.23%	88.99%	82.30%
	(59.2%)	(71.8%)	(71.4%)	(60.9%)	(73.3%)	(72.2%)	(3.3%)	(11.5%)	(16.6%)
BiTCN	1.93	2.83	3.78	3.36	4.52	5.95	95.77%	92.35%	86.73%
	(60.4%)	(67.4%)	(66.2%)	(63.0%)	(67.9%)	(67.9%)	(3.8%)	(7.4%)	(12.1%)
BiLSTM	1.47	1.76	1.95	2.47	2.83	3.11	97.72%	97.00%	96.38%
	(48.1%)	(47.7%)	(34.6%)	(49.6%)	(48.8%)	(38.5%)	(1.7%)	(2.3%)	(2.3%)
TCN-LSTM	1.27	2.07	2.06	2.33	3.20	3.42	98.00%	96.27%	95.64%
	(39.9%)	(55.4%)	(38.1%)	(46.6%)	(54.7%)	(44.0%)	(1.4%)	(3.1%)	(3.0%)
BiTCN-BiLSTM	1.21	1.58	1.89	2.24	3.07	3.38	98.12%	96.48%	95.86%
	(36.9%)	(41.7%)	(32.5%)	(44.5%)	(52.7%)	(43.4%)	(1.3%)	(2.8%)	(2.8%)
OTDBO-BiTCN-BiLSTM	1.13	1.40	1.88	1.95	2.97	3.08	98.57%	96.71%	96.45%
	(32.3%)	(34.3%)	(32.1%)	(36.4%)	(51.1%)	(37.9%)	(0.9%)	(2.6%)	(2.2%)
OTDBO-BiTCN-BiLSTM-Adaboost	1.09	1.32	1.87	1.86	2.66	3.02	98.70%	97.36%	96.59%
	(30.2%)	(30.2%)	(31.5%)	(33.3%)	(45.4%)	(36.7%)	(0.7%)	(1.9%)	(2.1%)
ISSA-OTDBO-BiTCN-BiLSTM	0.89	1.26	1.50	1.38	1.88	2.53	99.29%	98.68%	97.61%
	(13.7%)	(27.1%)	(14.8%)	(9.8%)	(22.8%)	(24.3%)	(0.1%)	(0.5%)	(1.03%)
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.76	0.92	1.28	1.24	1.45	1.91	99.42%	99.21%	98.63%

This table compares the performance of different forecasting models on the March dataset using multi-step-ahead predictions (1-step, 2-step, and 3-step).

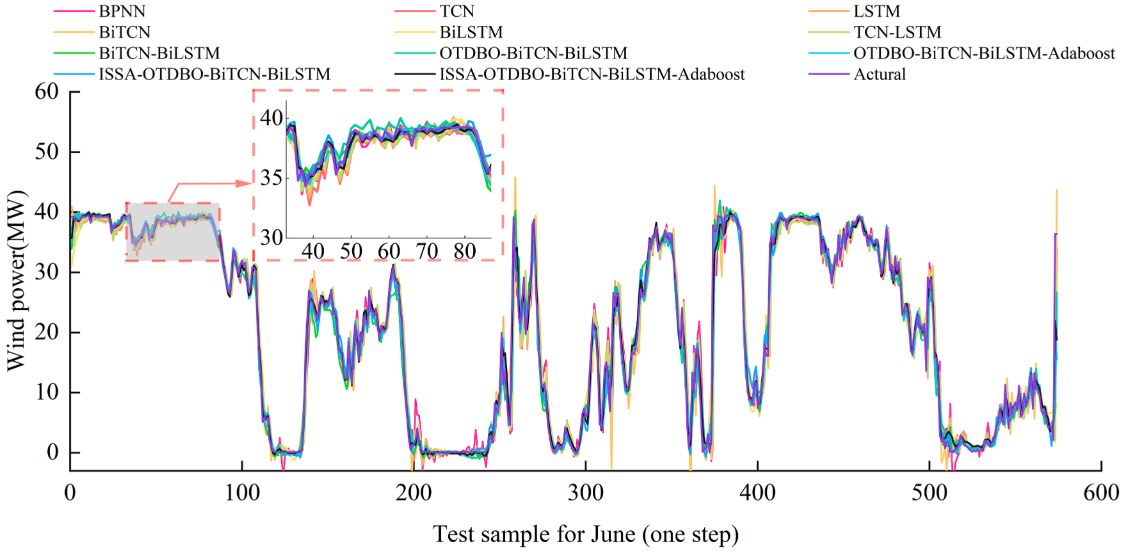


Figure 9. One-step-ahead prediction results for the June dataset

Forecasting performance of different models is evaluated using June test data. The inset provides a detailed view of a selected region.

dependency capture, outperforming unidirectional TCN-LSTM. Further, OTDBO optimization reduces MAE by 6.18% and increases R^2 by 0.42%, validating its effectiveness in dynamic hyperparameter tuning and feature learning precision.

Integrating Adaboost into the OTDBO-BiTCN-BiLSTM framework further reduces MAE by 3.26% and RMSE by 5.67%, while increasing R^2 by 0.31%. This is particularly beneficial for multi-step forecasting, mitigating error accumulation.

Table 5. Multi-step forecasting performance comparison (June dataset)

	June								
	MAE/IR$_{MAE}$			RMSE/IR$_{RMSE}$			R^2/IR$_{R^2}$		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BPNN	2.168	3.495	4.297	3.821	5.589	6.886	93.35%	85.79%	78.40%
	(65.8%)	(72.6%)	(68.4%)	(61.0%)	(67.4%)	(65.6%)	(5.7%)	(12.9%)	(19.5%)
TCN	2.121	3.227	4.062	3.760	5.517	6.621	93.52%	86.04%	79.87%
	(65.0%)	(70.3%)	(66.6%)	(60.4%)	(67.0%)	(64.2%)	(5.5%)	(12.6%)	(18.0%)
LSTM	2.102	3.221	4.548	3.779	5.524	7.018	93.49%	86.09%	77.55%
	(64.7%)	(70.2%)	(70.2%)	(60.6%)	(67.1%)	(66.2%)	(5.6%)	(12.6%)	(20.4%)
BiTCN	2.194	3.432	4.304	3.913	5.828	7.023	93.02%	84.52%	77.52%
	(66.2%)	(72.0%)	(68.5%)	(61.9%)	(68.8%)	(66.2%)	(6.0%)	(14.2%)	(20.4%)
BiLSTM	1.564	1.970	2.170	2.942	3.426	3.905	96.06%	94.66%	93.06%
	(52.5%)	(51.3%)	(37.5%)	(49.4%)	(46.9%)	(39.3%)	(3.0%)	(3.9%)	(4.5%)
TCN-LSTM	2.107	3.226	4.090	3.760	5.524	6.635	93.16%	84.83%	77.09%
	(64.8%)	(70.3%)	(66.8%)	(60.4%)	(67.1%)	(64.3%)	(5.9%)	(13.9%)	(20.9%)
BiTCN-BiLSTM	1.505	1.911	2.174	2.682	3.328	3.856	96.72%	94.95%	93.23%
	(50.7%)	(49.8%)	(37.6%)	(44.5%)	(45.3%)	(38.5%)	(2.3%)	(3.6%)	(4.3%)
OTDBO-BiTCN-BiLSTM	1.377	1.688	2.152	2.485	3.324	3.685	97.19%	94.97%	93.81%
	(46.1%)	(43.2%)	(37.0%)	(40.1%)	(45.2%)	(35.7%)	(1.8%)	(3.6%)	(3.7%)
OTDBO-BiTCN-BiLSTM-Adaboost	1.293	1.683	2.037	2.422	3.139	3.588	97.33%	95.51%	94.14%
	(42.6%)	(43.0%)	(33.4%)	(38.5%)	(42.0%)	(33.9%)	(1.7%)	(3.0%)	(3.4%)
ISSA-OTDBO-BiTCN-BiLSTM	0.915	1.105	1.560	1.700	2.008	2.826	98.68%	98.16%	96.36%
	(18.9%)	(13.2%)	(13.1%)	(12.4%)	(9.3%)	(16.1%)	(0.3%)	(0.3%)	(1.1%)
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.742	0.959	1.36	1.489	1.820	2.37	98.99%	98.49%	97.44%

This table presents the multi-step forecasting performance of different models on the June dataset, evaluating MAE, RMSE, and R^2 metrics.

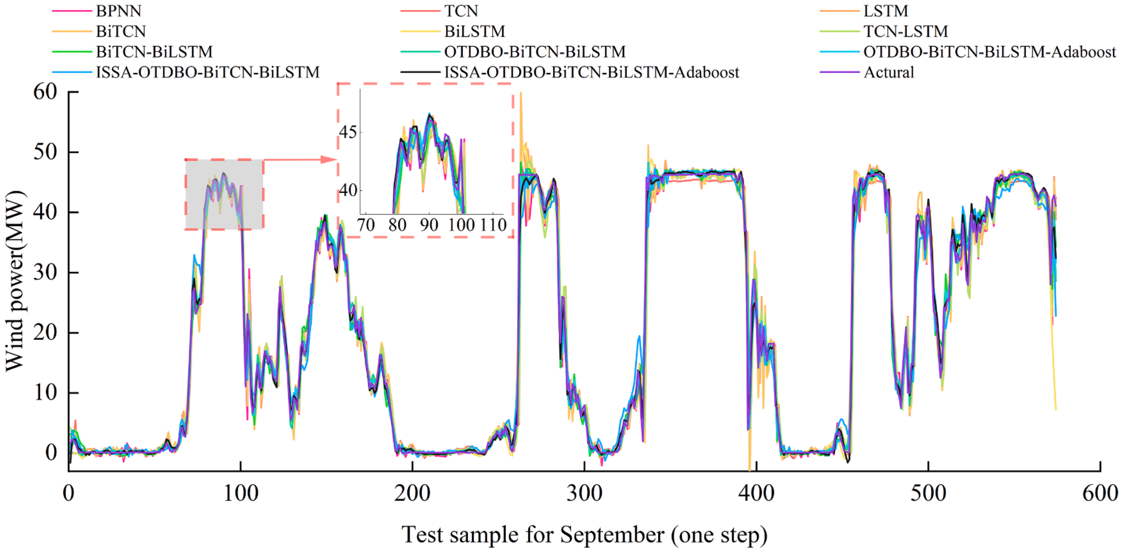


Figure 10. One-step-ahead prediction results for the September dataset

Model predictions for September data are compared against actual wind power output, with a magnified section showing finer details.

Finally, incorporating ISSA significantly enhances predictive performance, reducing MAE by 30.71% and RMSE by 38.53%, while improving R^2 by 1.54%. These results confirm ISSA's effectiveness in extracting key modal features, enhancing prediction stability and accuracy.

Figure 9 presents the one-step prediction results of 11 models on the June dataset, which includes 2880 data points, with 574 in the test set. Unlike the March dataset, which exhibits diurnal fluctuations, the June dataset shows consistently high wind power output with subtle variations due to increased summer wind speeds.

Table 6. Multi-step forecasting performance comparison (September dataset)

	September								
	MAE/IR_{MAE}			$RMSE/IR_{RMSE}$			R^2/IR_{R^2}		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BPNN	2.184	3.332	4.225	4.062	6.118	7.335	95.31%	89.37%	84.73%
	(68.1%)	(66.5%)	(69.9%)	(69.5%)	(72.8%)	(72.7%)	(4.3%)	(9.9%)	(14.3%)
TCN	2.049	3.175	4.051	4.062	6.110	7.403	95.31%	89.40%	84.45%
	(66.0%)	(64.9%)	(68.6%)	(69.5%)	(72.7%)	(72.9%)	(4.3%)	(9.8%)	(14.4%)
LSTM	2.063	3.419	4.981	4.099	6.227	7.940	95.23%	89.00%	82.18%
	(66.2%)	(67.4%)	(74.5%)	(69.8%)	(73.2%)	(74.8%)	(4.4%)	(10.3%)	(16.9%)
BiTCN	2.116	3.417	4.296	4.197	6.428	7.725	95.00%	88.27%	83.07%
	(67.0%)	(67.3%)	(70.4%)	(70.5%)	(74.1%)	(74.1%)	(4.6%)	(11.0%)	(16.0%)
BiLSTM	1.471	1.944	2.423	3.215	3.656	4.483	97.06%	96.20%	94.29%
	(52.6%)	(42.6%)	(47.6%)	(61.5%)	(54.4%)	(55.3%)	(2.5%)	(3.0%)	(4.6%)
TCN-LSTM	2.028	3.132	3.939	4.055	6.046	7.319	95.14%	88.92%	83.00%
	(65.6%)	(64.4%)	(67.7%)	(69.5%)	(72.4%)	(72.6%)	(4.4%)	(10.4%)	(16.0%)
BiTCN-BiLSTM	1.452	1.933	2.064	2.736	3.602	4.008	97.87%	96.32%	95.44%
	(52.0%)	(42.3%)	(38.4%)	(54.8%)	(53.7%)	(50.0%)	(1.7%)	(2.9%)	(3.5%)
OTDBO-BiTCN-BiLSTM	1.274	1.744	1.912	2.510	3.556	3.948	98.21%	96.41%	95.57%
	(45.2%)	(36.0%)	(33.5%)	(50.7%)	(53.1%)	(49.2%)	(1.4%)	(2.8%)	(3.3%)
OTDBO-BiTCN-BiLSTM-Adaboost	1.104	1.790	2.039	2.293	3.523	3.736	98.51%	96.48%	96.04%
	(36.8%)	(37.6%)	(37.7%)	(46.0%)	(52.7%)	(46.3%)	(1.1%)	(2.8%)	(2.9%)
ISSA-OTDBO-BiTCN-BiLSTM	0.906	1.162	1.286	1.505	2.117	2.191	99.36%	98.73%	98.64%
	(23.0%)	(3.9%)	(1.2%)	(17.8%)	(21.3%)	(8.5%)	(0.2%)	(0.5%)	(0.22%)
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.698	1.12	1.27	1.237	1.67	2.00	99.57%	99.21%	98.86%

This table provides a comparison of the forecasting accuracy of different models on the September dataset across multiple prediction steps.

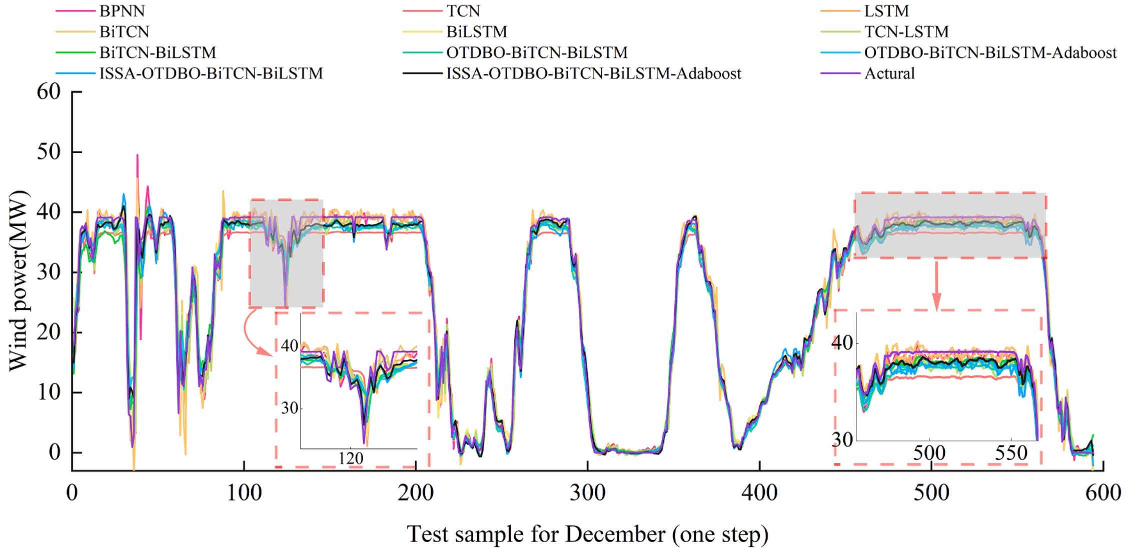


Figure 11. One-step-ahead prediction results for the December dataset
Performance comparison of multiple prediction models using December test data. The zoomed-in view emphasizes key differences.

These minor fluctuations pose a challenge for prediction models, yet the proposed model closely aligns with actual power values, demonstrating superior performance in tracking nuanced changes. By dynamically adjusting feature weights, it effectively captures seasonal patterns, showcasing strong adaptability and robustness. While other models capture short- and long-term

Table 7. Multi-step forecasting performance comparison (December dataset)

	December								
	MAE/IR$_{MAE}$			RMSE/IR$_{RMSE}$			R^2/IR$_R{}^2$		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BPNN	1.546	3.000	3.631	2.881	4.698	6.046	96.35%	90.29%	83.93%
	(45.5%)	(62.6%)	(68.7%)	(59.0%)	(66.4%)	(68.9%)	(3.1%)	(8.7%)	(14.7%)
TCN	2.200	3.062	4.052	3.165	4.487	5.744	95.65%	91.28%	85.74%
	(61.7%)	(63.3%)	(71.9%)	(62.7%)	(64.9%)	(67.2%)	(3.8%)	(7.7%)	(12.9%)
LSTM	1.533	2.601	4.250	2.836	4.410	5.964	96.46%	91.46%	84.64%
	(45.0%)	(56.8%)	(73.2%)	(58.4%)	(64.3%)	(68.4%)	(2.9%)	(7.5%)	(14.0%)
BiTCN	1.573	2.608	3.445	3.070	4.554	5.678	95.85%	90.89%	85.87%
	(46.4%)	(56.9%)	(67.0%)	(61.6%)	(65.4%)	(66.8%)	(3.6%)	(8.1%)	(12.8%)
BiLSTM	1.649	1.928	2.238	2.710	3.583	3.744	96.77%	94.35%	93.84%
	(48.9%)	(41.8%)	(49.2%)	(56.4%)	(56.0%)	(49.7%)	(2.6%)	(4.6%)	(4.7%)
TCN-LSTM	1.775	2.778	3.588	2.943	4.293	5.337	95.84%	90.59%	84.74%
	(52.5%)	(59.6%)	(68.3%)	(59.9%)	(63.3%)	(64.7%)	(3.6%)	(8.4%)	(13.9%)
BiTCN-BiLSTM	1.453	1.739	2.127	2.211	2.718	3.163	97.85%	96.62%	95.60%
	(42.0%)	(35.4%)	(46.5%)	(46.6%)	(42.0%)	(40.5%)	(1.5%)	(2.3%)	(2.9%)
OTDBO-BiTCN-BiLSTM	1.453	1.725	2.083	2.042	2.682	3.113	98.17%	96.75%	95.75%
	(41.9%)	(34.9%)	(45.4%)	(42.2%)	(41.2%)	(39.5%)	(1.2%)	(2.2%)	(2.7%)
OTDBO-BiTCN-BiLSTM-Adaboost	1.287	1.455	1.966	1.917	2.594	2.957	98.38%	97.04%	96.16%
	(34.5%)	(22.8%)	(42.1%)	(38.4%)	(39.2%)	(36.3%)	(1.0%)	(1.9%)	(2.3%)
ISSA-OTDBO-BiTCN-BiLSTM	1.203	1.204	1.366	1.616	1.743	2.035	98.85%	98.66%	98.18%
	(29.9%)	(6.7%)	(16.8%)	(26.9%)	(9.6%)	(7.5%)	(0.5%)	(0.2%)	(0.3%)
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.843	1.123	1.1371	1.180	1.576	1.8833	99.39%	98.91%	98.44%

This table provides a comparative analysis of multi-step forecasting performance for the December dataset, presenting key evaluation metrics such as MAE, RMSE, and R^2.

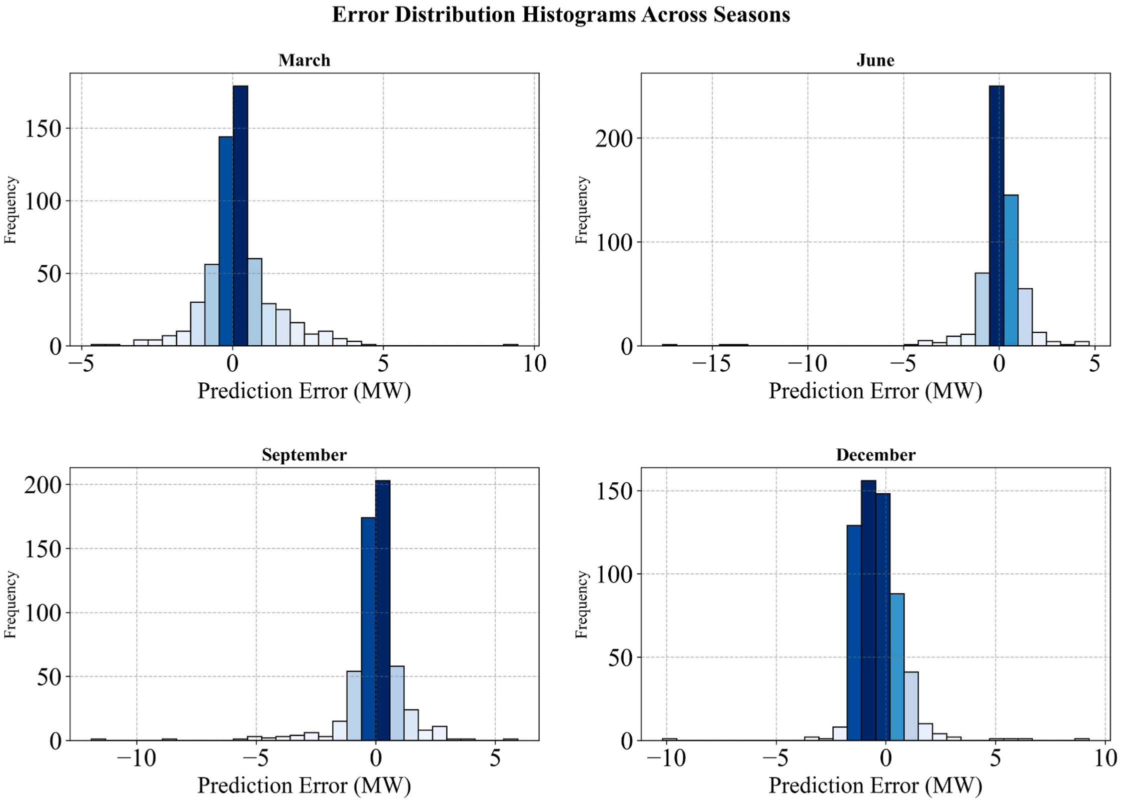


Figure 12. Error distribution of prediction results across seasons

Histograms of prediction errors for March, June, September, and December datasets, showing model performance variations.

dependencies to some extent, their accuracy is inferior, whereas the proposed model excels in learning structural patterns and responding to intricate variations in wind power data.

Table 5 summarizes the multi-step forecasting error analysis for all models on the June dataset. The BiTCN-BiLSTM baseline outperforms individual models across all metrics, showcasing strong feature extraction capabilities. OTDBO optimization further reduces MAE by up to 7.06% and improves R^2 by 0.36%. Integrating ISSA into OTDBO-BiTCN-BiLSTM significantly enhances performance, with MAE reductions of 28.99%, RMSE decreases of 29.03%, and R^2 improvements of 2.07%, demonstrating ISSA's effectiveness in capturing long-term trends while preserving local nuances. Adding Adaboost as an error correction layer further refines predictions, boosting R^2 to 98.99% for one-step forecasts and improving metrics across all steps. The ISSA-OTDBO-BiTCN-BiLSTM-Adaboost model achieves superior multi-step accuracy, excelling in capturing subtle variations and correcting errors, significantly outperforming all other models. This underscores its robustness and adaptability for complex time-series forecasting.

As shown in Figure 10, the September (autumn) wind power curve in Dabancheng contains more zero-power intervals than June, primarily due to reduced wind resources at lower temperatures. This causes wind speeds to fall below turbine cut-in thresholds, leading to shutdowns. The proposed model (black line) closely aligns with actual power values (purple line), accurately capturing power fluctuations and zero-power intervals. The September dataset includes complex fluctuations seen in March and June, along with multiple consecutive zero-power periods. Despite this, the proposed model achieves the lowest prediction error and best overall performance.

The zoomed-in region in Figure 10 further highlights the model's accuracy in predicting power declines and recoveries, while other models show larger errors near the cut-in threshold. This underscores the model's strong seasonal adaptability and ability to maintain high accuracy under varied meteorological conditions and power states.

Table 6 presents the multi-step forecasting error analysis for the September dataset, highlighting the proposed model's effectiveness. BiTCN-BiLSTM maintains one-step R^2 above 97%, demonstrating strong adaptability to seasonal variations. The optimized model outperforms the baseline, reducing multi-step MAE by 44.2%, RMSE by 52.8%, and improving R^2 by 2.7% on average.

Among enhancements, ISSA has the greatest impact, reducing MAE by 36.8%–37.7%, RMSE by 46%–52.7%, and improving R^2 by up to 2.9%. OTDBO further boosts accuracy through dynamic hyperparameter tuning, lowering MAE by 12.26%–7.36%, RMSE by 8.26%–1.50%, and increasing R^2 by 0.09%–0.34%. Adaboost significantly refines predictions, reducing MAE by up to 23%, RMSE by 21.3%, and improving R^2 by 0.5%.

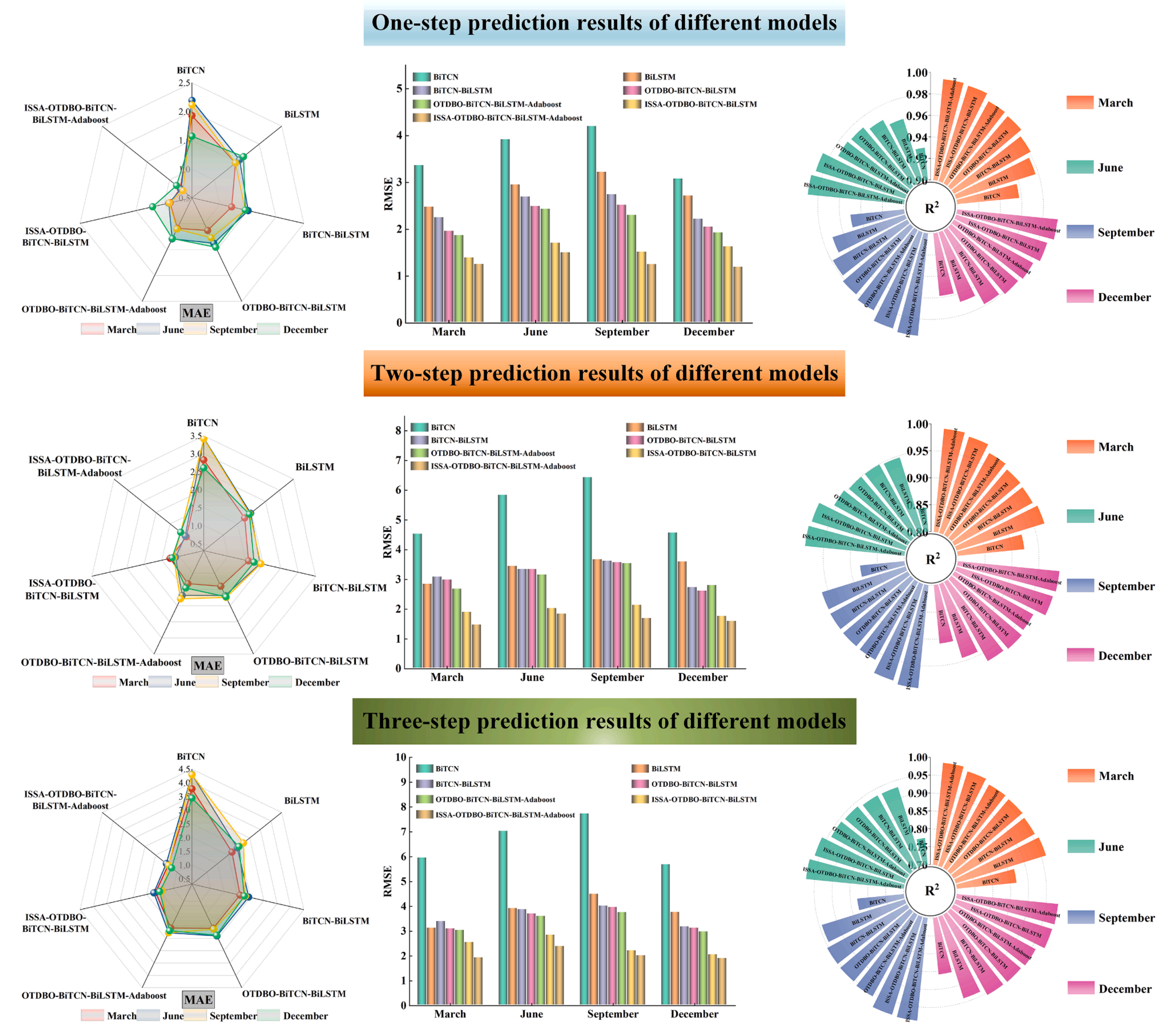


Figure 13. Statistical analysis of multi-step forecast results

Performance metrics (MAE, RMSE, R^2) for different models across one-step, two-step, and three-step predictions in four seasons.

Overall, ISSA enhances feature extraction, while OTDBO and Adaboost optimize model performance, generalization, and robustness.

As shown in Figure 11, all models perform well in capturing the overall trend of wind power variations, accurately predicting power fluctuations and peak occurrences. However, in terms of forecasting accuracy during abrupt power changes and peak regions, the proposed model demonstrates a significant advantage, with its predicted curve closely aligning with actual power values.

The two marked high-power intervals in the figure (samples 80–200 and 470–560) highlight the forecasting challenges posed by rare high-power patterns. Although all models exhibit relatively large errors in these regions, the proposed model remains capable of effectively tracking power fluctuation trends, resulting in comparatively lower prediction errors.

Table 7 presents the multi-step forecasting error analysis for the December dataset, highlighting key insights. The BiTCN-BiLSTM model performs similarly in December and September, achieving R^2 = 97.85%, RMSE = 2.211, and MAE = 1.453 for one-step forecasting. Integrating ISSA-OTDBO-BiTCN-BiLSTM-Adaboost significantly enhances accuracy, reducing MAE by 42%, 35.4%, and 45.6%, RMSE by 46.6%, 42%, and 40.5%, and improving R^2 by 1.5%, 2.3%, and 2.9% across multiple steps. ISSA's mode decomposition has the greatest impact, reducing MAE by 42.1%, RMSE by 39.2%, and increasing R^2 by up to 2.3%, aligning with its strong impact in earlier datasets.

Table 8. Comparative analysis of different decomposition methods

		MAE			RMSE			R^2		
Mths	Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
Mar	OTDBO-BiTCN-BiLSTM	1.13	1.40	1.88	1.95	2.97	3.08	98.57%	96.71%	96.45%
	ICEEMDAN-OTDBO-BiTCN-BiLSTM	1.31	2.08	2.38	1.96	2.87	3.53	98.56%	96.92%	95.35%
	TVF-EMD-OTDBO-BiTCN-BiLSTM	1.67	2.22	2.43	2.33	3.16	3.32	97.98%	96.28%	95.90%
	GA-VMD-OTDBO-BiTCN-BiLSTM	1.58	1.94	2.02	2.20	2.54	2.67	98.20%	97.58%	97.34%
	ISSA-OTDBO-BiTCN-BiLSTM	0.89	1.26	1.50	1.38	1.88	2.53	99.29%	98.68%	97.61%
Jun	OTDBO-BiTCN-BiLSTM	1.377	1.688	2.152	2.485	3.324	3.685	97.19%	94.97%	93.81%
	ICEEMDAN-OTDBO-BiTCN-BiLSTM	2.67	2.36	2.59	1.84	3.61	3.92	96.76%	94.05%	92.99%
	TVF-EMD-OTDBO-BiTCN-BiLSTM	3.73	4.07	4.05	5.06	5.52	5.56	88.67%	86.48%	86.28%
	GA-VMD-OTDBO-BiTCN-BiLSTM	1.7158	1.84	2.10	2.6221	2.84	2.98	96.87%	96.32%	95.94%
	ISSA-OTDBO-BiTCN-BiLSTM	0.915	1.105	1.560	1.700	2.008	2.826	98.68%	98.16%	96.36%
Sep	OTDBO-BiTCN-BiLSTM	1.274	1.790	2.039	2.510	3.556	3.948	98.21%	96.41%	95.57%
	ICEEMDAN-OTDBO-BiTCN-BiLSTM	2.00	2.11	2.31	2.96	3.09	3.49	97.52%	97.28%	96.55%
	TVF-EMD-OTDBO-BiTCN-BiLSTM	1.8652	2.33	2.82	2.7005	3.35	3.83	97.93%	96.82%	95.85%
	GA-VMD-OTDBO-BiTCN-BiLSTM	1.8035	2.23	2.1734	2.5559	2.97	3.0387	98.14%	97.50%	97.38%
	ISSA-OTDBO-BiTCN-BiLSTM	0.906	1.162	1.286	1.505	2.117	2.191	99.36%	98.73%	98.64%
Dec	OTDBO-BiTCN-BiLSTM	1.453	1.725	2.083	2.042	2.682	3.113	98.17%	96.75%	95.75%
	ICEEMDAN-OTDBO-BiTCN-BiLSTM	3.06	2.72	3.01	3.92	3.92	3.98	93.26%	93.24%	93.03%
	TVF-EMD-OTDBO-BiTCN-BiLSTM	4.08	4.14	4.46	4.56	4.57	4.96	91.32%	91.29%	89.94%
	GA-VMD-OTDBO-BiTCN-BiLSTM	1.5903	2.53	2.15	2.1702	1.77	3.00	97.93%	97.18%	96.06%
	ISSA-OTDBO-BiTCN-BiLSTM	1.203	1.204	1.366	1.616	1.743	2.035	98.85%	98.66%	98.18%

This table investigates the impact of different decomposition techniques on model performance, comparing methods such as ICEEMDAN, TVF-EMD, and GA-VMD against the proposed ISSA-OTDBO-BiTCN-BiLSTM approach.

The improvement from OTDBO is less pronounced in December than in March, June, and September, with MAE reduced by 2.1%, RMSE by 7.67%, and R^2 increased by 0.33%, likely due to BiTCN-BiLSTM's hyperparameters being near-optimal before tuning. Nonetheless, OTDBO remains effective. Adaboost, ranking second to ISSA, demonstrates strong adaptability to seasonal variations and consistent performance improvements across datasets.

To analyze the impact of seasonal variations on prediction errors, Figure 12 presents the forecasting error distribution across seasons. Most errors are centered around zero, confirming the model's high accuracy and stability, with minimal extreme errors, demonstrating strong generalization capability. However, variations in error distribution may result from wind speed fluctuations, monsoon effects, or localized meteorological conditions.

For a comprehensive evaluation, Figure 13 presents radar charts, bar plots, and radial bar charts of multi-step forecasting errors across four seasons. As the forecasting horizon increases, MAE rises significantly, reflecting the weakened temporal dependencies in multi-step forecasting, making it harder to capture long-term variations. Notably, MAE, RMSE, and R^2 vary across seasons, indicating that wind power forecasting performance is influenced by seasonal characteristics. However, the proposed ISSA-OTDBO-BiTCN-BiLSTM-Adaboost model exhibits the smallest error fluctuations and maintains consistently high R^2 values, demonstrating superior adaptability and stability under varying wind conditions.

Comparison of the impact of different model decomposition methods on prediction results

This section evaluates the effects of ISSA and other decomposition methods (ICEEMDAN, TVF-EMD, GA-VMD) on model performance across four seasonal datasets.

Table 8 and Figure 14 show that the ISSA-OTDBO-BiTCN-BiLSTM model achieves the highest prediction accuracy across all seasons, demonstrating superior MAE, RMSE, and R^2 values for 1-step, 2-step, and 3-step predictions. This is attributed to ISSA's ability to perform multi-scale feature decomposition, reduce noise, and capture peak mutations. Conversely, ICEEMDAN and TVF-EMD failed to improve or even worsened prediction accuracy due to added noise or loss of key information during decomposition.

GA-VMD performs better than ICEEMDAN and TVF-EMD but is still less effective than ISSA, particularly in complex seasonal conditions. Additionally, 1-step predictions showed the lowest MAE and RMSE across all decomposition methods, with prediction accuracy declining as the prediction step length increased. This reflects increased uncertainty in longer forecasts. The findings highlight ISSA's robustness and superior ability to manage complex wind power time series, emphasizing its practical value for accurate forecasting.

Comparison of the impact of different optimization algorithms on prediction performance

To evaluate the applicability and robustness of optimization algorithms in short-term wind power forecasting, this section

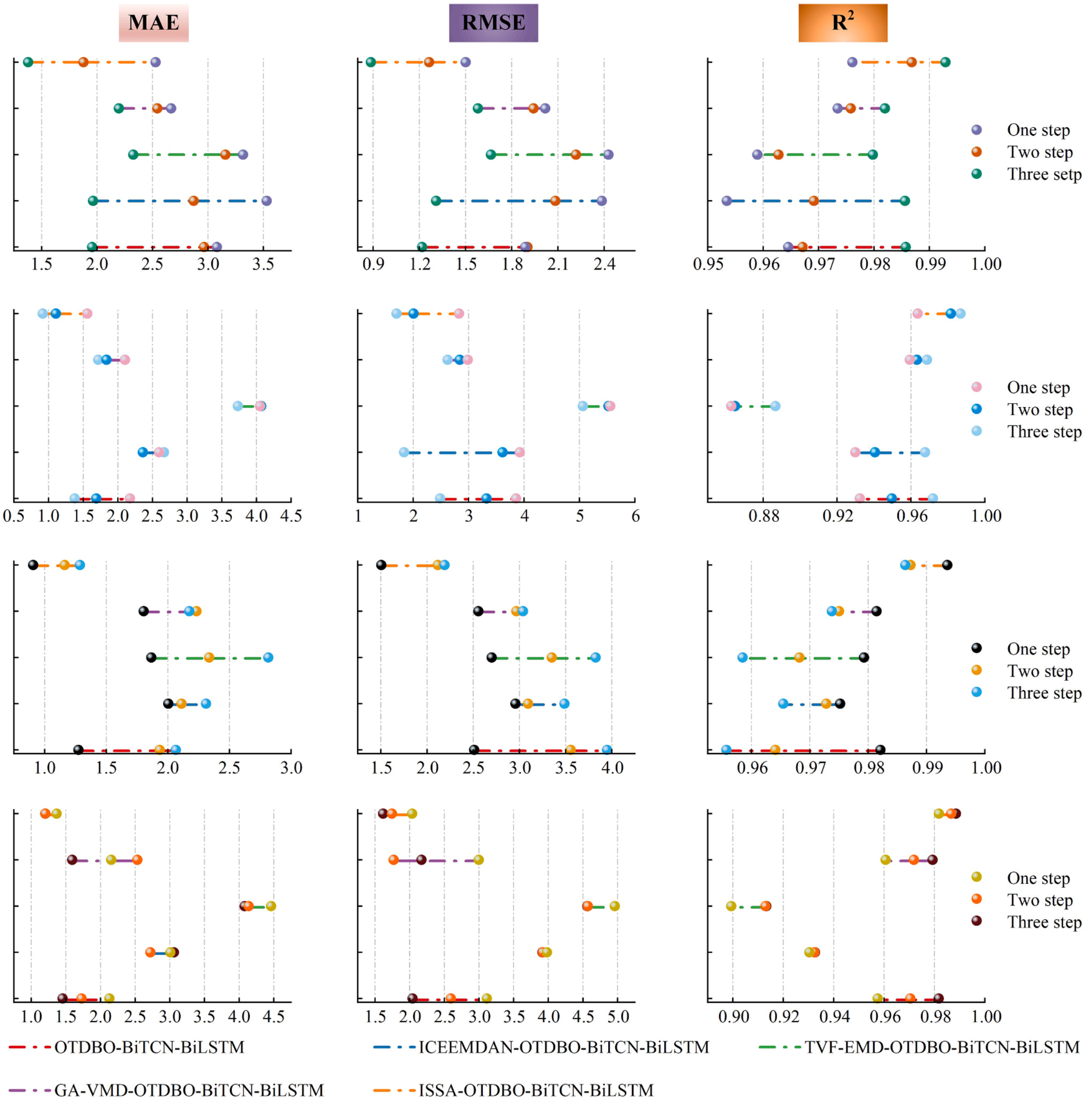


Figure 14. Prediction error statistics of different decomposition methods

MAE, RMSE, and R^2 comparisons for models using different signal decomposition techniques.

conducts a comparative analysis of three optimization algorithms—WOA, DBO, and OTDBO—within the proposed ISSA-BiTCN-BiLSTM-Adaboost model framework. The study comprehensively assesses the global search capability, convergence speed, and optimization stability of each algorithm.

Table 9 presents the error performance of different optimization algorithms across four seasons, while Figure 15 illustrates their convergence curves and scatterplots for one-step forecasting. Results show that OTDBO consistently outperforms WOA and DBO in MAE, RMSE, and R^2, demonstrating superior stability and generalization, especially in multi-step forecasting.

In March, OTDBO achieves the lowest errors (1-step MAE: 0.76, RMSE: 1.24) and maintains R^2 above 99%, outperforming WOA in accuracy and speed while DBO struggles with instability in deeper temporal dependencies.

In June, OTDBO's robustness is evident, achieving a 1-step MAE of 0.742, much lower than WOA (1.21) and DBO (1.52), with the lowest RMSE (1.489). WOA and DBO exhibit instability, with 2-step errors exceeding 3-step errors, while OTDBO maintains stable accuracy.

In September, as forecasting horizons increase, OTDBO remains precise (3-step MAE: 1.27), while WOA and DBO show significant error escalation (3.88 and 3.15, respectively). OTDBO retains an R^2 of 98.86%, whereas WOA drops to 92.29%, demonstrating superior adaptation to seasonal variability.

In December, OTDBO leads across all metrics (3-step MAE: 1.137, R^2: 99.39%), handling complex winter wind conditions more effectively. Faster convergence and stronger global search capabilities further enhance accuracy.

Table 9. Impact of optimization algorithms on prediction accuracy

		MAE/IR_{MA}			$RMSE/IR_{RMSE}$			R^2/IR_{R^2}		
Mths	Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
Mar	ISSA-WOA-BiTCN-BiLSTM-Adaboost	1.09 (29.7%)	1.73 (46.7%)	1.76 (27.2%)	1.53 (18.8%)	2.24 (35.1%)	2.65 (27.7%)	99.12% (0.3%)	98.14% (1.1%)	97.38% (1.3%)
	ISSA-DBO-BiTCN-BiLSTM-Adaboost	0.94 (18.5%)	1.13 (18.7%)	1.48 (13.6%)	1.30 (4.3%)	1.79 (18.9%)	2.19 (12.5%)	99.37% (0.1%)	98.80% (0.4%)	98.21% (0.4%)
	ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.76	0.92	1.28	1.24	1.45	1.91	99.42%	99.21%	98.63%
Jun	ISSA-WOA-BiTCN-BiLSTM-Adaboost	1.21 (38.8%)	1.88 (49.1%)	1.7051 (20.5%)	2.16 (31.1%)	2.98 (38.9%)	2.6728 (11.3%)	97.87% (1.1%)	95.97% (2.6%)	96.74% (0.7%)
	ISSA-DBO-BiTCN-BiLSTM-Adaboost	1.52 (51.3%)	1.80 (46.8%)	1.84 (26.2%)	2.64 (43.6%)	2.99 (39.1%)	2.93 (19.0%)	96.84% (2.2%)	95.93% (2.6%)	96.09% (1.4%)
	ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.742	0.959	1.36	1.489	1.820	2.37	98.99%	98.49%	97.44%
Sep	ISSA-WOA-BiTCN-BiLSTM-Adaboost	2.32 (70.0%)	1.74 (35.9%)	3.88 (67.2%)	3.54 (65.0%)	3.20 (47.9%)	5.23 (61.7%)	96.46% (3.1%)	97.09% (2.1%)	92.29% (6.6%)
	ISSA-DBO-BiTCN-BiLSTM-Adaboost	1.14 (38.7%)	2.08 (46.3%)	3.1483 (59.6%)	1.75 (29.2%)	2.72 (38.6%)	4.8967 (59.1%)	99.13% (0.4%)	97.91% (1.3%)	93.20% (5.7%)
	ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.698	1.12	1.27	1.237	1.67	2.00	99.57%	99.21%	98.86%
Dec	ISSA-WOA-BiTCN-BiLSTM-Adaboost	0.90 (6.1%)	1.27 (11.8%)	1.62 (29.9%)	1.30 (9.0%)	1.95 (19.1%)	2.33 (19.1%)	99.26% (0.1%)	98.33% (0.6%)	97.62% (0.8%)
	ISSA-DBO-BiTCN-BiLSTM-Adaboost	1.09 (23.0%)	1.30 (13.3%)	1.24 (8.4%)	1.58 (25.2%)	1.95 (19.2%)	2.04 (7.9%)	98.90% (0.5%)	98.33% (0.6%)	98.16% (0.3%)
	ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.843	1.123	1.1371	1.180	1.576	1.8833	99.39%	98.91%	98.44%

This table evaluates the effectiveness of various optimization algorithms in improving prediction accuracy, including WOA, DBO, and OTDBO, analyzing their impact on MAE, RMSE, and R^2.

Overall, OTDBO achieves lower errors, better stability, and superior optimization, while WOA and DBO suffer from convergence and generalization limitations, especially in multi-step forecasting. These findings confirm OTDBO's strong global search ability and predictive reliability, making it the optimal choice for wind power forecasting.

Comparison and analysis of ensemble learning Adaboost on improving prediction performance

This section evaluates the error correction capability of Adaboost and its impact on prediction accuracy by comparing its performance enhancement across different forecasting models.

Table 10 shows that Adaboost enhances all forecasting models across 1-step, 2-step, and 3-step predictions, though the degree of improvement varies. For BiTCN-BiLSTM, 1-step MAE decreases from 1.21 to 1.08 (a 10.7% reduction), while RMSE drops from 2.24 to 1.90 (a 15% reduction), indicating partial error correction. The impact is even greater for ISSA-OTDBO-BiTCN-BiLSTM, where 1-step MAE falls from 0.89 to 0.76, and R^2 improves from 99.29% to 99.42%, demonstrating Adaboost's effectiveness in refining errors even in high-accuracy models.

However, improvements are minimal in some models. For instance, CNN-BiLSTM sees only a slight RMSE reduction from 3.48 to 3.40 in 1-step forecasting, with negligible multi-step gains. Similarly, SVM shows marginal adjustments, suggesting Adaboost offers limited benefits for simpler deep models and traditional machine learning approaches.

Table 11 shows that all forecasting models improve after integrating Adaboost in the June dataset, with BiTCN-BiLSTM-Adaboost and SVM-Adaboost excelling in multi-step forecasting.

For BiTCN-BiLSTM, 1-step MAE drops from 1.505 to 1.21, while R^2 improves from 96.72% to 97.68%, confirming Adaboost's effectiveness in enhancing accuracy, even in highly fluctuating data. ISSA-OTDBO-BiTCN-BiLSTM-Adaboost maintains its superiority, with 3-step MAE decreasing from 1.560 to 1.36, RMSE from 2.826 to 2.37, and R^2 rising from 96.36% to 97.44%. Adaboost significantly enhances robustness and noise resistance in 2-step and 3-step forecasting, reducing error accumulation in complex time-series data.

In contrast, CNN-BiLSTM and Transformer models show only marginal R^2 improvements in multi-step forecasting, suggesting limited adaptability to non-stationary wind power data, with Adaboost providing minimal benefit.

Table 12 shows that all base models improve after integrating Adaboost in the September dataset, with varying degrees of error reduction and accuracy enhancement.

The ISSA-OTDBO-BiTCN-BiLSTM model remains the top performer, with 1-step MAE dropping from 0.906 to 0.698, RMSE decreasing from 1.505 to 1.237, and R^2 rising from 99.36% to 99.57%. This confirms Adaboost's ability to enhance noise resistance and accuracy, even under intensified autumn wind fluctuations.

Notably, SVM-Adaboost achieves a 3-step R^2 increase of 0.69 percentage points, reaching 94.51%, demonstrating that Adaboost benefits both deep networks and traditional machine learning models.

In contrast, while CNN-BiLSTM and Transformer models show minor MAE and RMSE improvements, their R^2 remains nearly

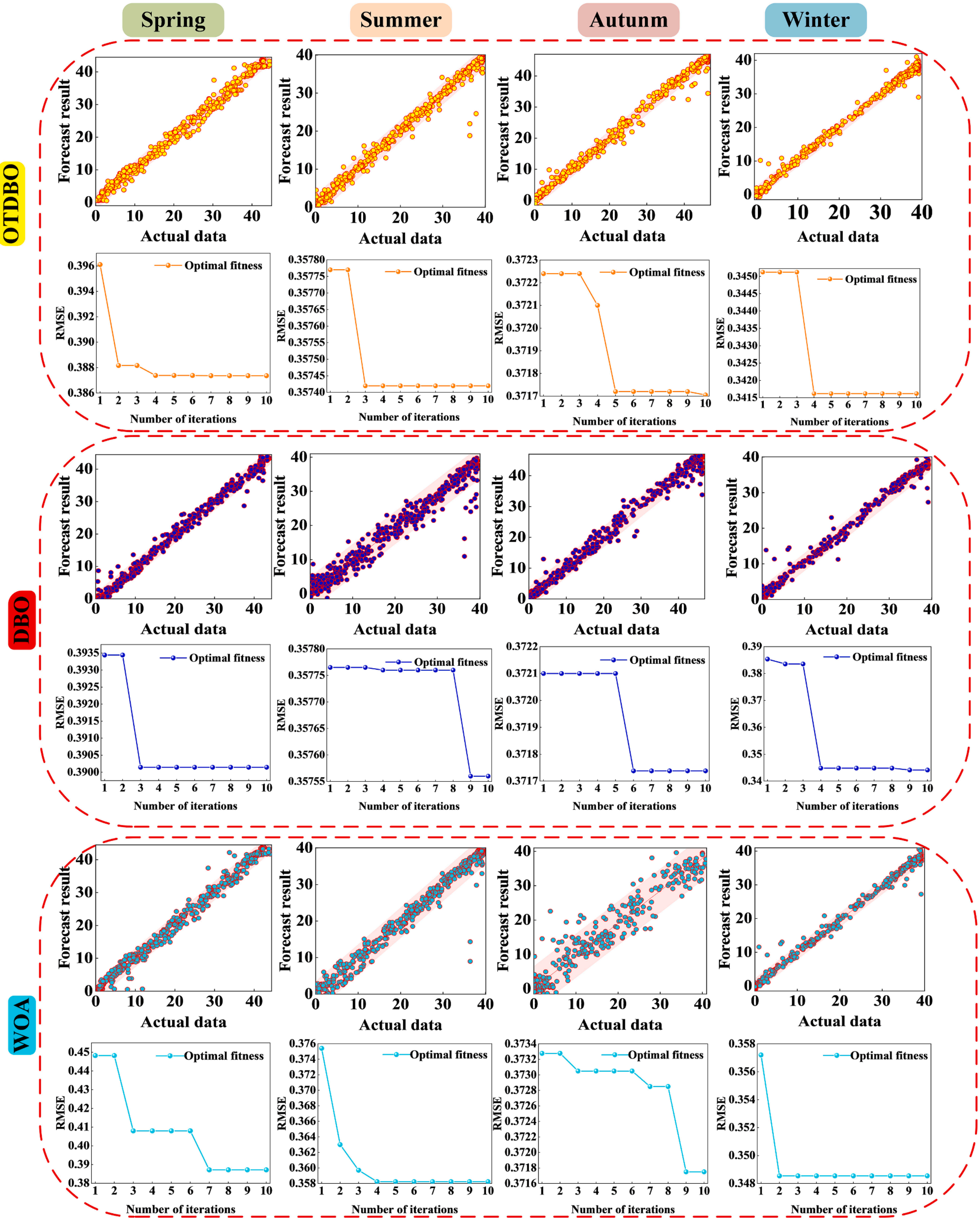


Figure 15. Iteration curves and scatterplots of optimization algorithms

Forecast results and optimization convergence curves for OTDBO, DBO, and WOA across four seasonal datasets.

Table 10. Performance enhancement of Adaboost on prediction models (March dataset)

	March								
	MAE			RMSE			R^2		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BiTCN-BiLSTM	1.21	1.58	1.89	2.24	3.07	3.38	98.12%	96.48%	95.86%
BiTCN-BiLSTM-Adaboost	1.08	1.93	1.93	1.90	3.02	3.17	98.65%	96.58%	96.25%
CNN-BiGRU	1.65	2.96	3.81	2.39	4.48	5.69	97.87%	92.50%	87.90%
CNN-BiGRU-Adaboost	1.46	2.66	3.53	2.17	4.08	5.45	98.25%	93.78%	88.90%
CNN-BiLSTM	2.02	3.06	4.09	3.48	4.99	6.37	95.48%	90.70%	84.81%
CNN-BiLSTM-Adaboost	1.88	2.97	3.97	3.40	4.98	6.25	95.68%	90.72%	85.40%
Transformer	1.90	3.49	3.99	3.18	5.12	6.31	96.21%	90.29%	85.12%
transformer-Adaboost	1.87	3.17	4.11	3.15	4.97	6.28	96.30%	90.81%	85.27%
SVM	1.44	2.28	2.87	2.68	3.89	4.55	97.31%	94.30%	92.23%
SVM-Adaboost	1.43	2.28	2.87	2.67	3.90	4.56	97.33%	94.31%	92.22%
ISSA-OTDBO-BiTCN-BiLSTM	0.89	1.26	1.50	1.38	1.88	2.53	99.29%	98.68%	97.61%
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.76	0.92	1.28	1.24	1.45	1.91	99.42%	99.21%	98.63%

This table presents a comparison of Adaboost-enhanced prediction models for the March dataset, assessing the improvement in forecasting accuracy through MAE, RMSE, and R^2 scores.

unchanged, indicating limited adaptability to seasonal variations. This suggests that Adaboost alone cannot fully compensate for their architectural constraints.

Table 13 shows that all base models improve after integrating Adaboost, particularly in multi-step forecasting, where MAE and RMSE reductions are more pronounced.

The ISSA-OTDBO-BiTCN-BiLSTM-Adaboost model further leads in 3-step forecasting, with MAE dropping to 1.1371, RMSE to 1.8833, and R^2 rising to 98.44%, demonstrating high adaptability and robustness in complex time-series forecasting. Notably, the Transformer model exhibits a significant 3-step RMSE reduction (from 5.84 to 5.22) and an R^2 increase (from 85.39% to 88.04%), highlighting Adaboost's role in enhancing stability under high-fluctuation winter conditions.

Although BiTCN-BiLSTM and CNN-BiGRU achieve higher R^2 in certain forecasting horizons, their overall performance remains inferior to ISSA-OTDBO-BiTCN-BiLSTM. Meanwhile, CNN-BiLSTM experiences increased errors, with the 1-step MAE rising (from 1.95 to 2.02) and the 3-step R^2 slightly declining (from 86.46% to 86.39%), suggesting that Adaboost's weight adjustments may not align well with certain architectures and data distributions, potentially leading to higher errors.

Overall, ISSA-OTDBO-BiTCN-BiLSTM-Adaboost achieves near-optimal accuracy across seasons and forecasting horizons, validating its efficiency and stability in multi-step prediction. To visually demonstrate Adaboost's impact, Figure 16 presents a bar comparison chart of 1-step predictions across four months, with color-coded models and background markers highlighting

Table 11. Performance enhancement of Adaboost on prediction models (June dataset)

	June								
	MAE/IR_{MAE}			$RMSE/IR_{RMSE}$			R^2/IR_{R}^2		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BiTCN-BiLSTM	1.505	1.911	2.174	2.682	3.328	3.856	96.72%	94.95%	93.23%
BiTCN-BiLSTM-Adaboost	1.21	1.56	2.03	2.26	3.02	3.50	97.68%	95.86%	94.41%
CNN-BiGRU	1.89	2.90	4.08	3.06	4.73	6.22	95.76%	89.85%	82.39%
CNN-BiGRU-Adaboost	1.53	2.79	3.97	2.34	4.57	6.21	97.50%	90.51%	82.46%
CNN-BiLSTM	2.21	3.33	4.12	3.88	5.58	6.64	93.15%	85.84%	79.92%
CNN-BiLSTM-Adaboost	2.20	3.28	4.09	3.87	5.55	6.61	93.19%	85.99%	80.08%
Transformer	2.03	3.28	3.98	3.70	5.46	6.58	93.77%	86.46%	80.33%
transformer-Adaboost	2.03	3.27	3.92	3.69	5.42	6.62	93.80%	86.61%	80.06%
SVM	1.48	2.08	2.46	2.62	3.51	3.92	96.87%	94.36%	92.96%
SVM-Adaboost	1.40	1.99	2.37	2.42	3.29	3.66	97.31%	95.04%	93.86%
ISSA-OTDBO-BiTCN-BiLSTM	0.915	1.105	1.560	1.700	2.008	2.826	98.68%	98.16%	96.36%
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.742	0.959	1.36	1.489	1.820	2.37	98.99%	98.49%	97.44%

This table examines the impact of Adaboost on different prediction models using the June dataset, demonstrating performance gains in one-step, two-step, and three-step forecasting scenarios.

Table 12. Performance enhancement of Adaboost on prediction models (September dataset)

	September								
	MAE/IR_{MAE}			$RMSE/IR_{RMSE}$			R^2/IR_{R^2}		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BiTCN-BiLSTM	1.452	1.933	2.064	2.736	3.602	4.008	97.87%	96.32%	95.44%
BiTCN-BiLSTM-Adaboost	1.19	1.75	2.11	2.31	3.22	3.86	98.48%	97.06%	95.77%
CNN-BiGRU	1.91	3.00	4.10	3.13	5.15	6.75	97.22%	92.50%	87.06%
CNN-BiGRU-Adaboost	1.69	2.94	4.07	2.77	5.02	6.79	97.83%	92.85%	86.93%
CNN-BiLSTM	2.29	3.27	4.05	4.27	6.11	7.36	94.83%	89.41%	84.66%
CNN-BiLSTM-Adaboost	2.17	3.23	4.09	4.19	6.12	7.38	95.03%	89.37%	84.57%
Transformer	1.98	3.63	4.52	4.03	6.24	7.59	95.38%	89.03%	83.86%
transformer-Adaboost	1.9677	3.23	4.23	3.9788	6.05	7.47	95.50%	89.61%	84.26%
SVM	1.62	2.44	2.93	3.00	4.06	4.49	97.39%	95.08%	93.82%
SVM-Adaboost	1.57	2.36	2.88	2.77	3.81	4.25	97.79%	95.69%	94.51%
ISSA-OTDBO-BiTCN-BiLSTM	0.906	1.162	1.286	1.505	2.117	2.191	99.36%	98.73%	98.64%
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.698	1.12	1.27	1.237	1.67	2.00	99.57%	99.21%	98.86%

This table evaluates the role of Adaboost in improving prediction accuracy for the September dataset, comparing its performance across different baseline models.

performance changes before and after Adaboost integration. This further confirms Adaboost's potential while emphasizing the need for careful assessment of its applicability in real-world scenarios.

Model interpretability and statistical analysis

To improve model interpretability, the SHAP (Shapley additive explanations) method is used to analyze the contribution of wind-related features to forecasting results. Given the ensemble nature of BiTCN-BiLSTM, a weighted average of SHAP values across all sub-models is computed to determine overall feature importance.

Figure 17 presents the SHAP analysis results: the SHAP summary plot (left) illustrates how feature variations impact predictions, while the SHAP feature importance plot (right) shows the average influence of each variable on model outputs.

Results confirm that wind speed is the most critical factor, followed by wind direction and air density. This aligns with the previous feature selection analysis, further validating the chosen input variables.

To assess the statistical significance of the proposed model's prediction accuracy improvements, a one-tailed paired t test was conducted across four seasons, comparing the proposed model to baseline models using MAE, RMSE, and R^2. The null hypothesis (H_0) states that no significant difference exists in prediction errors, while the alternative hypothesis (H_1) asserts that the proposed model significantly reduces errors.

Table 14 shows that all *p* values are below 0.001, with MAE reductions 34.2%–65.6%, RMSE reductions 38.1%–71.3%, and R^2 improvements 0.5 to 7.8 percentage points, confirming that the performance gains are statistically significant.

To assess computational efficiency, training and inference times were measured for baseline models and models of similar complexity using the March dataset (Table 15). The proposed

Table 13. Comparison of Adaboost's performance enhancement on prediction models (December dataset)

	December								
	MAE/IR_{MAE}			$RMSE/IR_{RMSE}$			R^2/IR_{R^2}		
Model	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
BiTCN-BiLSTM	1.453	1.739	2.127	2.211	2.718	3.163	97.85%	96.62%	95.60%
BiTCN-BiLSTM-Adaboost	1.37	1.74	1.95	2.10	2.68	3.00	98.06%	96.84%	96.04%
CNN-BiGRU	1.64	2.24	2.91	2.27	3.75	4.75	97.73%	93.84%	90.07%
CNN-BiGRU-Adaboost	1.54	2.23	2.86	2.06	3.54	4.81	98.14%	94.49%	89.82%
CNN-BiLSTM	1.95	2.84	3.77	3.12	4.50	5.57	95.71%	91.14%	86.46%
CNN-BiLSTM-Adaboost	2.02	2.95	3.77	3.16	4.51	5.58	95.61%	91.09%	86.39%
Transformer	1.60	2.73	4.38	2.85	4.33	5.84	96.44%	91.79%	85.39%
transformer-Adaboost	1.45	2.94	3.37	2.79	4.36	5.22	96.57%	91.72%	88.04%
SVM	1.28	1.97	2.48	2.17	3.07	3.66	97.93%	95.89%	94.17%
SVM-Adaboost	1.25	1.97	2.45	2.06	2.96	3.52	98.10%	96.03%	94.38%
ISSA-OTDBO-BiTCN-BiLSTM	1.203	1.204	1.366	1.616	1.743	2.035	98.85%	98.66%	98.18%
ISSA-OTDBO-BiTCN-BiLSTM-Adaboost	0.843	1.123	1.1371	1.180	1.576	1.8833	99.39%	98.91%	98.44%

This table evaluates the impact of Adaboost on different prediction models for the December dataset, considering MAE, RMSE, and R^2.

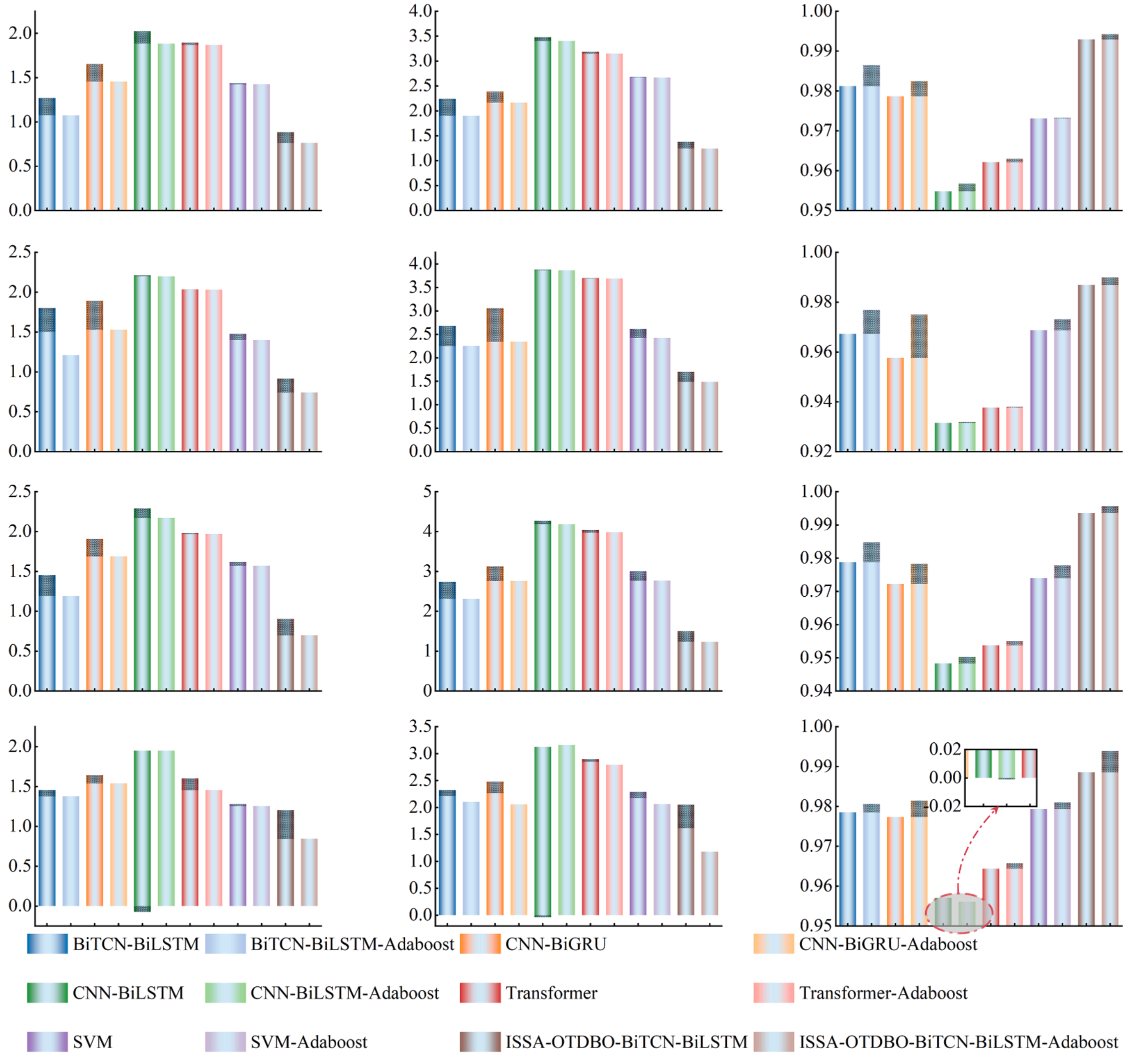


Figure 16. Adaboost's improvement in prediction performance

Comparative analysis of Adaboost-enhanced models versus baseline models using MAE, RMSE, and R^2.

model's training time increases from 54.28s (BiLSTM) to 3751.53s, but achieves a 49.7% RMSE reduction, justifying the complexity increase for higher accuracy.

Among comparable models, the proposed model requires 2946.85 s for training, which is 21.3% and 12.1% shorter than ISSA-WOA-BiTCN-BiLSTM-Adaboost (3743.08 s) and

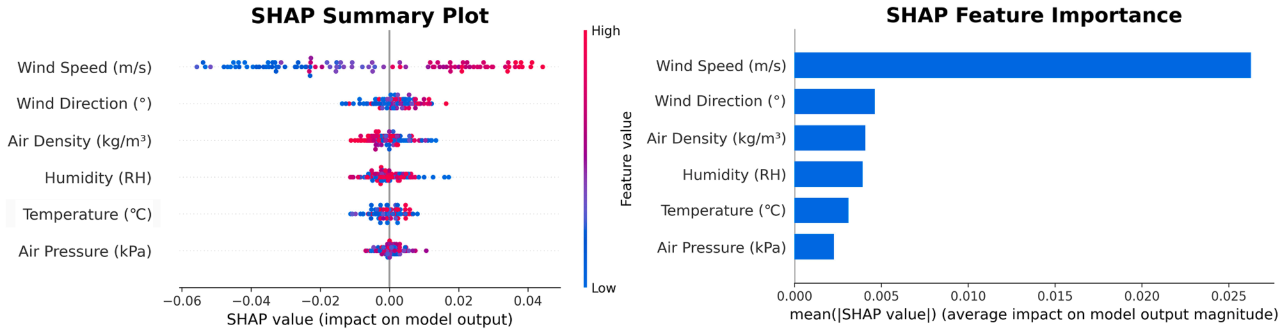


Figure 17. Feature contribution and importance ranking based on SHAP analysis

SHAP values indicate the impact of meteorological variables on wind power prediction.

Table 14. Paired t test analysis of the proposed model and other models

Comparison Model vs. Proposed Model	*p* value (MAE)	Significance (MAE)	*p* value (RMSE)	Significance (RMSE)	*p* value (R^2)	Significance (R^2)
BPNN	0.0032	✓	0.0025	✓	0.0018	✓
TCN	<0.0001	✓	<0.0001	✓	<0.0001	✓
LSTM	0.0007	✓	0.0004	✓	0.0002	✓
BiTCN	0.0011	✓	0.0009	✓	0.0006	✓
BiLSTM	0.0048	✓	0.0039	✓	0.0021	✓
TCN-LSTM	0.0002	✓	<0.0001	✓	<0.0001	✓
BiTCN-BiLSTM	0.0019	✓	0.0015	✓	0.0009	✓
OTDBO-BiTCN-BiLSTM	0.0024	✓	0.0018	✓	0.0010	✓
OTDBO-BiTCN-BiLSTM-Adaboost	0.0036	✓	0.0027	✓	0.0015	✓

This table presents the statistical significance of performance differences between the proposed model and other baseline models using paired t tests.

ISSA-DBO-BiTCN-BiLSTM-Adaboost (3351.48 s), while attaining the lowest RMSE (1.24). These results highlight OTDBO's dual advantage in accelerating convergence and reducing prediction errors.

Conclusion

This study addresses the challenges of low accuracy and instability in wind power forecasting caused by the nonlinear, stochastic, and multi-scale characteristics of wind power data. A novel deep hybrid forecasting model, ISSA-OTDBO-BiTCN-BiLSTM-Adaboost, is proposed, integrating ISSA, an enhanced OTDBO algorithm, and Adaboost ensemble learning to enhance prediction accuracy, robustness, and generalization. The model's effectiveness is validated using four seasonal datasets from the Dabancheng Wind Farm in Xinjiang. Furthermore, SHAP interpretability analysis and paired t-tests confirm its superiority. The main findings are as follows.

1. Across all seasons and forecast horizons, ISSA-OTDBO-BiTCN-BiLSTM-Adaboost achieves the lowest MAE and RM SE while attaining the highest R^2, demonstrating superior overall predictive accuracy.
2. Data decomposition significantly enhances wind power forecasting accuracy. Compared to ICEEMDAN, TVF-EMD, and GA-VMD, ISSA achieves average MAE reductions of 42.8%, 51.5%, and 41.5%, and RMSE reductions of 35.5%, 48.2%, and 32.9%, respectively.
3. OTDBO-based hyperparameter optimization significantly improves forecasting performance. Compared to WOA and DBO, OTDBO reduces MAE by 36.15% and 32.88%, respectively, while increasing R^2 by 1.15%, indicating that the enhanced OTDBO algorithm optimally fine-tunes model parameters.
4. Adaboost effectively integrates with deep learning models, further correcting prediction errors and mitigating error accumulation. Its correction efficacy depends on the deep learning model architecture. In the ISSA-OTDBO-BiTCN-BiLSTM framework, Adaboost reduces MAE by 14.3%, RMSE by 19.7%, and improves R^2 by up to 1.02%.

Table 15. Computational efficiency comparison of different models

Module	Training time(s)	Inference Time(s)	RMSE
BiLSTM	54.28	0.03	2.94
BiTCN	18.83	0.09	3.91
Temporal fusion transformer	450.20	0.06	2.10
Informer	620.55	0.08	2.05
CNN-BiGRU	82.42	0.02	3.06
CNN-BiGRU-Adaboost	338.49	0.07	2.34
BiTCN-BiLSTM	60.46	0.03	2.68
OTDBO- BiTCN-BiLSTM	1314.42	0.03	1.95
WOA-BiTCN-BiLSTM	2008.53	0.03	2.25
DBO-BiTCN-BiLSTM	1676.93	0.04	2.69
BiTCN-BiLSTM-Adaboost	313.51	0.08	2.42
ISSA- BiTCN-BiLSTM	301.23	0.05	1.70
ISSA- WOA- BiTCN-BiLSTM-Adaboost	3743.08	0.21	1.53
ISSA- DBO- BiTCN-BiLSTM-Adaboost	3351.48	0.21	1.30
ISSA- OTDBO- BiTCN-BiLSTM-Adaboost	2946.85	0.21	1.24

This table compares the computational efficiency of various models in terms of training time, inference time, and RMSE.

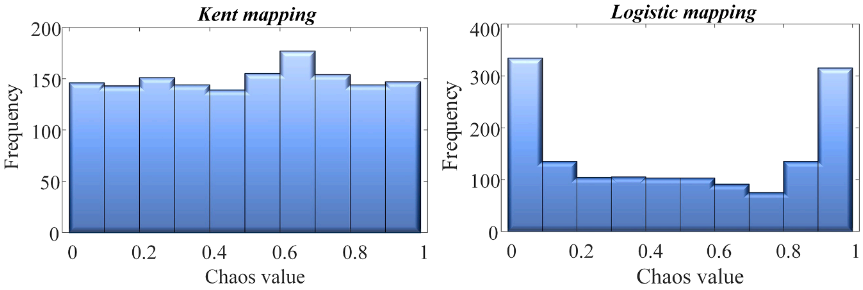


Figure 18. Frequency distribution histogram of Kent and logistic chaotic mapping

This figure shows the frequency distribution of chaos values generated by Kent mapping (left) and logistic mapping (right). The x axis represents the chaos values, while the y axis denotes the frequency of occurrence. The Kent mapping exhibits a more uniform distribution, whereas the logistic mapping shows concentration at the extremes (0 and 1), indicating different dynamical behaviors. These distributions highlight the impact of different chaotic mappings on subsequent data processing and modeling.

5. SHAP analysis shows that wind speed contributes over 40% to SHAP values across all forecast horizons, followed by wind direction, validating input feature selection. Paired t tests confirm statistical significance, with all p values below 0.05, proving the model's superiority over baselines.
6. Achieving high-accuracy predictions with hybrid models requires balancing accuracy and computational cost. Under similar complexity, the proposed model reduces training time by at least 12.1%, validating its efficiency-accuracy trade-off.

Limitations of the study

The proposed model exhibits strong predictive performance and holds practical application potential. Future work can focus on enhancing computational efficiency by reducing model complexity, exploring lightweight ensemble methods, and leveraging incremental learning to optimize training. Additionally, designing lightweight deep learning architectures will improve deployment feasibility on edge computing platforms. Parallel, distributed, and cloud computing can further enhance scalability for large-scale applications.

To broaden applicability, future research could extend the model to power load forecasting and hybrid wind-solar prediction. Combining physics-based forecasting with data-driven approaches could enhance robustness under extreme weather, optimizing multi-energy system dispatch. IoT-based wind monitoring could enable real-time adaptive forecasting, improving responsiveness to sudden wind fluctuations.

Further research should assess generalization across different regions and wind patterns to enhance practicality and reliability. Additionally, multi-objective optimization can balance forecasting accuracy, computational cost, and energy dispatch, increasing the model's economic value and real-world impact.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and information should be directed to and will be fulfilled by Prof. Lixin Zhang (zhlx2001329@163.com).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All data reported in this paper will be shared by the lead contact upon request.
- All original code has been deposited at Github and is publicly available as of the date of publication. DOI is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

List of nomenclature

ISSA	Improved singular spectrum analysis	CNN	Convolutional neural network
BiTCN	Bi-directional temporal convolutional network	BiGRU	Bi-directional gated recurrent unit
Adaboost	Adaptive boosting	SVM	Support vector machine
OOA	Osprey optimization algorithm	RF	Random forest
WOA	Whale optimization algorithm	MAE	Mean absolute error
DBO	Dung beetle optimizer	RMSE	Root-mean-square error
DBSCAN	Density-based spatial clustering of applications with noise	R^2	Goodness of fit
TCN	Temporal convolutional network	SCADA	Supervisory control and data acquisition
BiLSTM	Bi-directional long short-term memory	NWP	Numerical weather report
BPNN	Back propagation neural network	ICEEMDAN	Complete ensemble empirical mode decomposition with adaptive noise
EMD	Empirical mode decomposition	$IR_R{}^2$	Improvement rate of R2
GA	Genetic algorithm	IR_{MAE}	Improvement rate of MAE
VMD	Variational mode decomposition	IR_{RMSE}	Improvement rate of RMSE

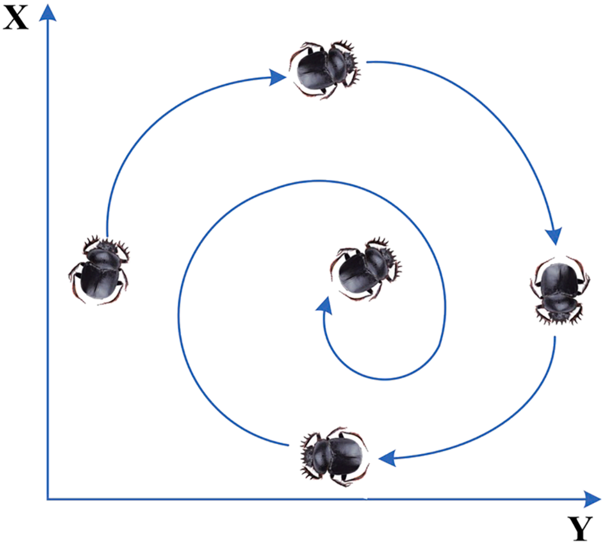


Figure 19. Search path of dung beetles during the breeding stage

This schematic illustrates the spiral search path of dung beetles during their breeding stage. The beetles navigate along a logarithmic spiral trajectory, which mimics natural foraging behavior. The X and Y axes represent the search space dimensions. This pattern is the basis for the dung beetle optimization (DBO) algorithm, where the search behavior is adapted for hyperparameter tuning in machine learning models.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (grant no. 52406267) and the Major Science and Technology Project of the Autonomous Region (grant no. 2022A01004-4).

AUTHOR CONTRIBUTIONS

For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used conceptualization, C.M. and X.C.; methodology, C.M.; software, C.M. and L.Z.; validation, C.M.; formal analysis, C.M. and X.C; investigation, C.M. and L.Z.; data curation, C.M. and X.H.; writing—original draft preparation, C.M. and O.B.; writing—review and editing, C.M. and O.B.; visualization, C.M.; supervision, C.M.; project administration, L.Z. and X.H.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

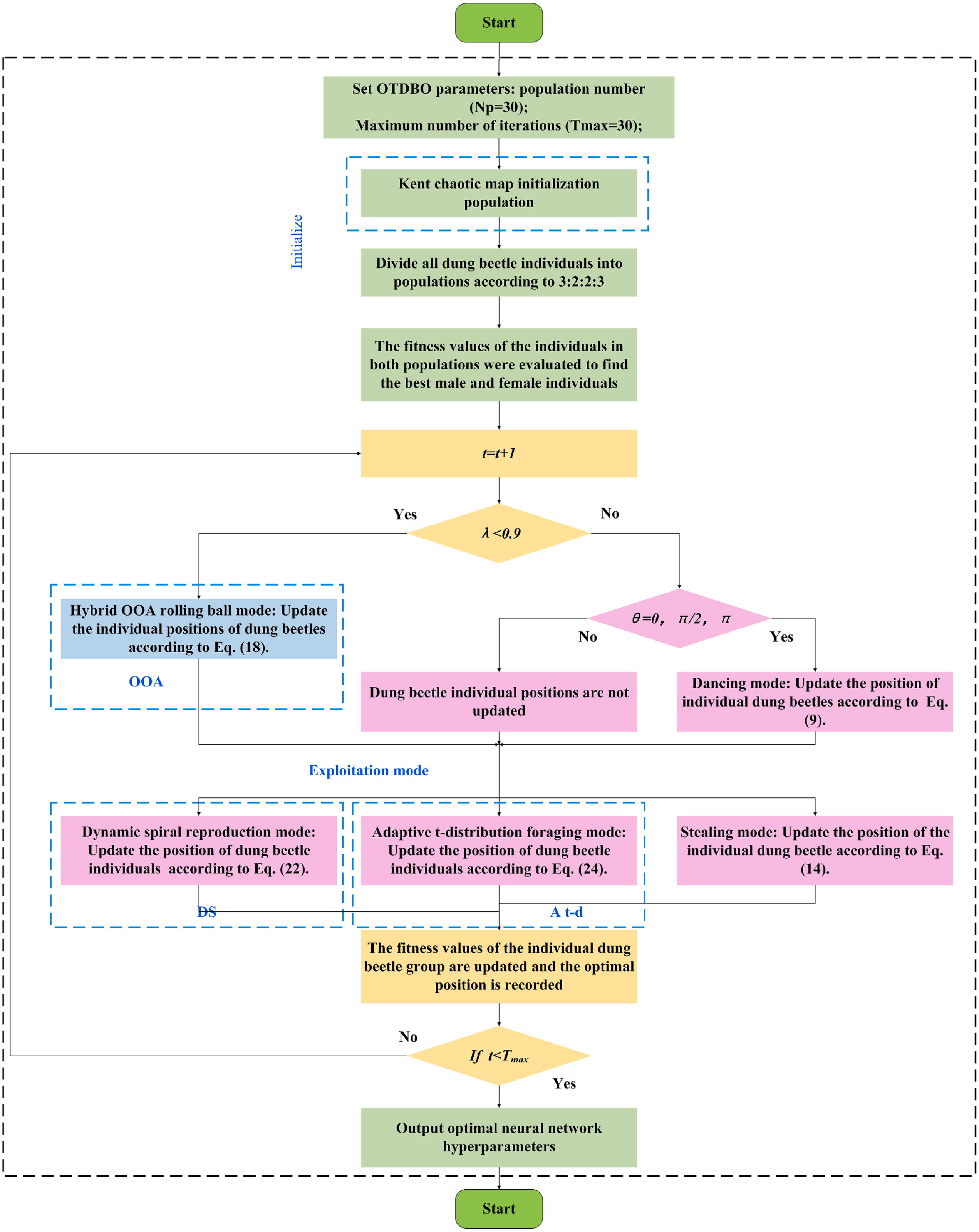
Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
  - Adaptive singular spectrum analysis mode decomposition based on autocorrelation function
  - Matrix decomposition
  - Decomposition
  - Grouping
  - Reconstruction
  - Adaptive dimension selection
  - Dung beetle optimization algorithm
  - Rolling dung beetles
  - Breeding dung beetles
  - Small dung beetle (foraging dung beetle)
  - Stealing dung beetle
  - Improved dung beetle optimization algorithm
- QUANTIFICATION AND STATISTICAL ANALYSIS

Received: January 18, 2025
Revised: February 25, 2025
Accepted: April 2, 2025
Published: April 6, 2025

REFERENCES

1. Rahman, A., Farrok, O., and Haque, M.M. (2022). Environmental impact of renewable energy source based electrical power plants: Solar, wind, hydroelectric, biomass, geothermal, tidal, ocean, and osmotic. Renew. Sustain. Energy Rev. *161*, 112279. https://doi.org/10.1016/j.rser.2022.112279.
2. Liu, K., Chen, W., Chen, G., Dai, D., Ai, C., Zhang, X., and Wang, X. (2023). Application and analysis of hydraulic wind power generation technology. Energy Strategy Rev. *48*, 101117. https://doi.org/10.1016/j.esr.2023.101117.
3. Wang, Y., Zou, R., Liu, F., Zhang, L., and Liu, Q. (2021). A review of wind speed and wind power forecasting with deep neural networks. Appl. Energy *304*, 117766. https://doi.org/10.1016/j.apenergy.2021.117766.
4. Ahmed, M., Gamal, M., Ismail, I.,M.E., and El-Din, H. (2024). An AI-Based System for Predicting Renewable Energy Power Output Using Advanced Optimization Algorithms. J. Artif. Intell. Metaheuristics *8*, 1–8. https://doi.org/10.54216/JAIM.080101.
5. Tuncar, E.A., Saglam, S., and Oral, B. (2024). A review of short-term wind power generation forecasting methods in recent technological trends. Energy Rep. *12*, 197–209. https://doi.org/10.1016/j.egyr.2024.06.006.
6. Kirchner-Bossi, N., Kathari, G., and Porté-Agel, F. (2024). A hybrid physics-based and data-driven model for intra-day and day-ahead wind power forecasting considering a drastically expanded predictor search space. Appl. Energy *367*, 123375. https://doi.org/10.1016/j.apenergy.2024.123375.
7. González Sopeña, J.M., Pakrashi, V., and Ghosh, B. (2023). A benchmarking framework for performance evaluation of statistical wind power forecasting models. Sustain. Energy Technol. Assessments *57*, 103246. https://doi.org/10.1016/j.seta.2023.103246.
8. Kim, Y., and Hur, J. (2020). An ensemble forecasting model of wind power outputs based on improved statistical approaches. Energies *13*, 1071. https://doi.org/10.3390/en13051071.
9. Tawn, R., and Browell, J. (2022). A review of very short-term wind and solar power forecasting. Renew. Sustain. Energy Rev. *153*, 111758. https://doi.org/10.1016/j.rser.2021.111758.
10. Hassan, Q., Algburi, S., Sameen, A.Z., Salman, H.M., and Jaszczur, M. (2023). A review of hybrid renewable energy systems: Solar and wind-powered solutions: Challenges, opportunities, and policy implications. Results Eng. *20*, 101621. https://doi.org/10.1016/j.rinene.2023.101621.
11. Sfetsos, A. (2000). A comparison of various forecasting techniques applied to mean hourly wind speed time series. Renew. Energy *21*, 23–35. https://doi.org/10.1016/S0960-1481(99)00125-1.
12. Singh, S.N., Singh, S.n., and Mohapatra, A. (2019). Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. Renew. Energy *136*, 758–768. https://doi.org/10.1016/j.renene.2019.01.031.
13. Tena García, J.L., Cadenas Calderón, E., González Ávalos, G., Rangel Heras, E., and Mbikayi Tshikala, A. (2019). Forecast of daily output energy of wind turbine using SARIMA and nonlinear autoregressive models. Adv. Mech. Eng. *11*, 1687814018813464. https://doi.org/10.1177/1687814018813464.


Start
Set OTDBO parameters: population number (Np=30);
Maximum number of iterations (Tmax=30);
Kent chaotic map initialization population
Initialize
Divide all dung beetle individuals into populations according to 3:2:2:3
The fitness values of the individuals in both populations were evaluated to find the best male and female individuals
t=t+1
Yes
λ <0.9
No
Hybrid OOA rolling ball mode: Update the individual positions of dung beetles according to Eq. (18).
OOA
θ=0, π/2, π
No
Yes
Dung beetle individual positions are not updated
Dancing mode: Update the position of individual dung beetles according to Eq. (9).
Exploitation mode
Dynamic spiral reproduction mode: Update the position of dung beetle individuals according to Eq. (22).
Adaptive t-distribution foraging mode: Update the position of dung beetle individuals according to Eq. (24).
Stealing mode: Update the position of the individual dung beetle according to Eq. (14).
DS
A t-d
The fitness values of the individual dung beetle group are updated and the optimal position is recorded
No
If t<T_max
Yes
Output optimal neural network hyperparameters
Start

(legend on next page)

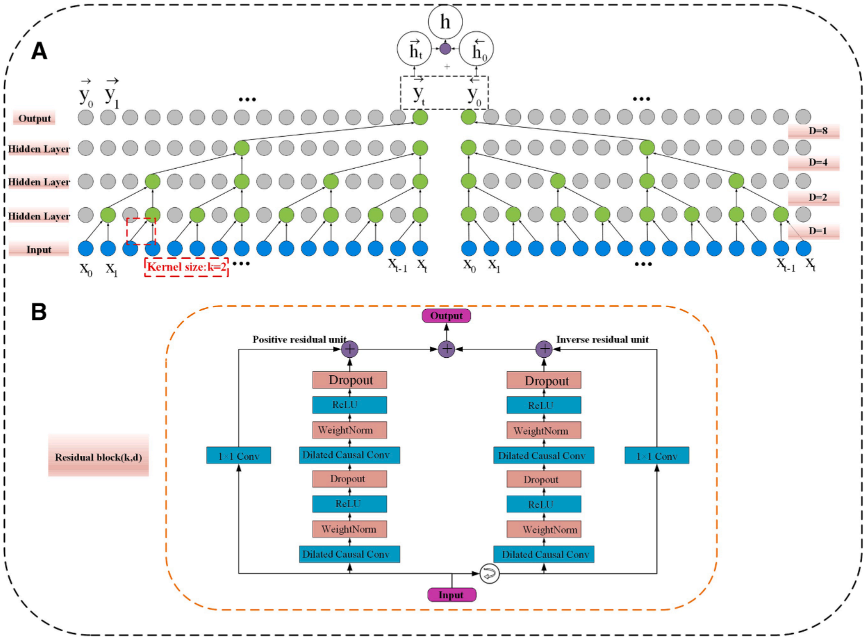


Figure 21. Architecture of the BiTCN Network

(A) BiTCN structure with bidirectional temporal convolutional layers and kernel size k = 2k = 2k = 2.

(B) Residual block design with dilated causal convolutions, dropout, ReLU, and weight normalization.

14. Li, Y., Su, Y., and Shu, L. (2014). An ARMAX model for forecasting the power output of a grid connected photovoltaic system. Renew. Energy *66*, 78–89. https://doi.org/10.1016/j.renene.2013.11.067.

15. Anh, N.T.N., Anh, N.N., Thang, T.N., Solanki, V.K., Crespo, R.G., and Dat, N.Q. (2024). Online SARIMA applied for short-term electricity load forecasting. Appl. Intell. *54*, 1003–1019. https://doi.org/10.1007/s10489-023-05230-y.

16. Yunus, K., Thiringer, T., and Chen, P. (2016). ARIMA-based frequency-decomposed modeling of wind speed time series. IEEE Trans. Power Syst. *31*, 2546–2556. https://doi.org/10.1109/TPWRS.2015.2468586.

17. Biswas, A.K., Ahmed, S.I., Bankefa, T., Ranganathan, P., and Salehfar, H. (2021). Performance analysis of short and mid-term wind power prediction using ARIMA and hybrid models. In 2021 IEEE Power and Energy Conference at Illinois (PECI) (IEEE), pp. 1–7. https://doi.org/10.1109/TPEC.

18. Demolli, H., Dokuz, A.S., Ecemis, A., and Gokcek, M. (2019). Wind power forecasting based on daily wind speed data using machine learning algorithms. Energy Convers. Manag. *198*, 111823. https://doi.org/10.1016/j.enconman.2019.111823.

19. Ponkumar, G., Jayaprakash, S., and Kanagarathinam, K. (2023). Advanced machine learning techniques for accurate very-short-term wind power forecasting in wind energy systems using historical data analysis. Energies *16*, 5459. https://doi.org/10.3390/en16145459.

20. M, E.,M.,M., and Abualigah, L. (2024). Machine Learning in Public Health Forecasting and Monitoring the Zika Virus. Metaheuristic Optim. Rev. *1*, 1–11. https://doi.org/10.54216/MOR.010201.

21. Eed, M., Alhussan, A.A., Qenawy, A.S.T., Osman, A.M., Elshewey, A.M., and Arnous, R. (2025). Potato consumption forecasting based on a hybrid stacked deep learning model. Potato Res. *68*, 809–833. https://doi.org/10.1007/s11540-024-09764-7.

22. Wan, C., Xu, Z., Pinson, P., Dong, Z.Y., and Wong, K.P. (2014). Probabilistic forecasting of wind power generation using extreme learning machine. IEEE Trans. Power Syst. *29*, 1033–1044. https://doi.org/10.1109/TPWRS.2013.2287876.

23. Yang, L., He, M., Zhang, J., and Vittal, V. (2015). Support-vector-machine-enhanced Markov model for short-term wind power forecast. IEEE Trans. Sustain. Energy *6*, 791–799. https://doi.org/10.1109/TSTE.2015.2406814.

24. Sun, W., Liu, M., and Liang, Y. (2015). Wind speed forecasting based on FEEMD and LSSVM optimized by the bat algorithm. Energies *8*, 6585–6607. https://doi.org/10.3390/en8076585.

25. Chen, M.R., Zeng, G.Q., Lu, K.D., and Weng, J. (2019). A two-layer nonlinear combination method for short-term wind speed prediction based on ELM, ENN, and LSTM. IEEE Internet Things J. *6*, 6997–7010. https://doi.org/10.1109/JIOT.2019.2913176.

26. Qu, Z., Mao, W., Zhang, K., Zhang, W., and Li, Z. (2019). Multi-step wind speed forecasting based on a hybrid decomposition technique and an improved back-propagation neural network. Renew. Energy *133*, 919–929. https://doi.org/10.1016/j.renene.2018.10.043.

27. Wang, J., Yang, W., Du, P., and Niu, T. (2018). A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm. Energy Convers. Manag. *163*, 134–150. https://doi.org/10.1016/j.enconman.2018.02.012.

28. Ding, J., Chen, G., Huang, Y., Zhu, Z., Yuan, K., and Xu, H. (2021). Short-term wind speed prediction based on CEEMDAN-SE-improved PIO-GRNN model. Meas. Control *54*, 73–87. https://doi.org/10.1177/0020294020981400.

29. Yuan, X., Chen, C., Yuan, Y., Huang, Y., and Tan, Q. (2015). Short-term wind power prediction based on LSSVM–GSA model. Energy Convers. Manag. *101*, 393–401. https://doi.org/10.1016/j.enconman.2015.05.063.

Figure 20. Flowchart of the OTDBO algorithm

The OTDBO algorithm optimizes neural network hyperparameters through iterative updates. It initializes the population using the Kent chaotic map, followed by population division and fitness evaluation. The search process alternates between OOA rolling ball, dynamic spiral reproduction (DS), adaptive t-distribution foraging (A t-d), dancing, and stealing modes. When the maximum iteration condition is met, the optimal hyperparameters are output. Different colors indicate distinct operation stages.

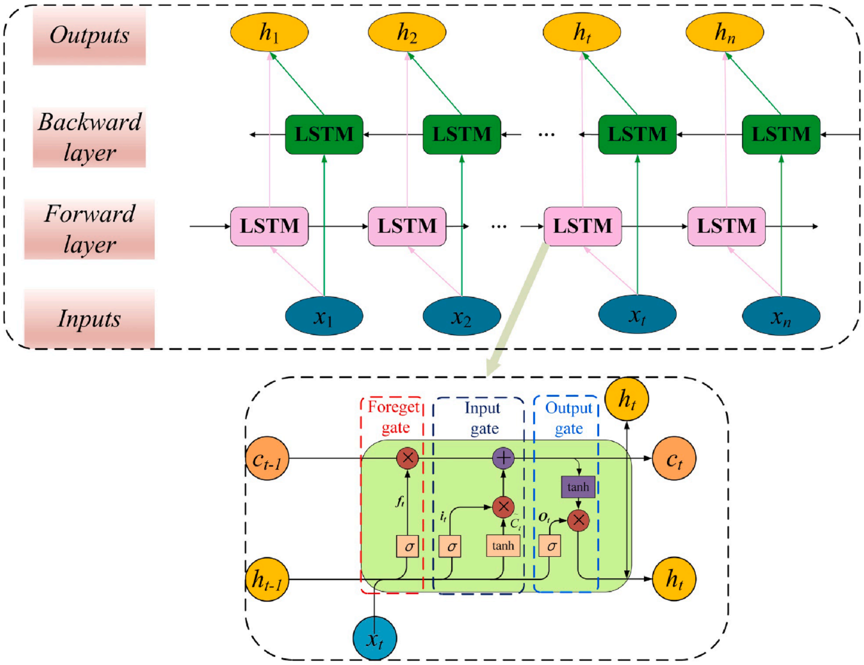


Figure 22. Architecture of the BiLSTM NetworkThe BiLSTM network consists of forward and backward LSTM layers to capture bidirectional dependencies

The lower panel illustrates the internal structure of an LSTM unit, including the forget, input, and output gates.

30. Li, L.L., Liu, Z.F., Tseng, M.L., Jantarakolica, K., and Lim, M.K. (2021). Using enhanced crow search algorithm optimization-extreme learning machine model to forecast short-term wind power. Expert Syst. Appl. *184*, 115579. https://doi.org/10.1016/j.eswa.2021.115579.

31. Wu, Z., and Wang, B. (2021). An ensemble neural network based on variational mode decomposition and an improved sparrow search algorithm for wind and solar power forecasting. IEEE Access *9*, 166709–166719. https://doi.org/10.1109/ACCESS.2021.3136387.

32. Shi, Z., Liang, H., and Dinavahi, V. (2018). Direct interval forecast of uncertain wind power based on recurrent neural networks. IEEE Trans. Sustain. Energy *9*, 1177–1187. https://doi.org/10.1109/TSTE.2017.2778503.

33. Liu, X., and Zhou, J. (2024). Short-term wind power forecasting based on multivariate/multi-step LSTM with temporal feature attention mechanism. Appl. Soft Comput. *150*, 111050. https://doi.org/10.1016/j.asoc.2023.111050.

34. Fantini, D.G., Silva, R.N., Siqueira, M.B.B., Pinto, M.S.S., Guimarães, M., and Brasil, A.c.p., Junior. (2024). Wind speed short-term prediction using recurrent neural network GRU model and stationary wavelet transform GRU hybrid model. Energy Convers. Manag. *308*, 118333. https://doi.org/10.1016/j.enconman.2024.118333.

35. Lawal, A., Rehman, S., Alhems, L.M., and Alam, M.M. (2021). Wind speed prediction using hybrid 1D CNN and BLSTM network. IEEE Access *9*, 156672–156679. https://doi.org/10.1109/ACCESS.2021.3129883.

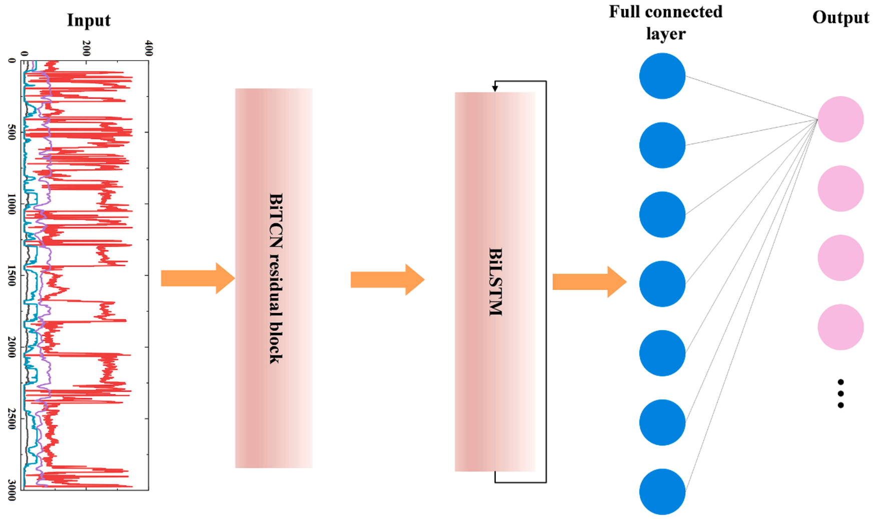


Figure 23. Architecture of the BiTCN-BiLSTM Model

The model processes input time-series data using BiTCN residual blocks, followed by BiLSTM layers for temporal feature extraction, and a fully connected layer for final output generation.

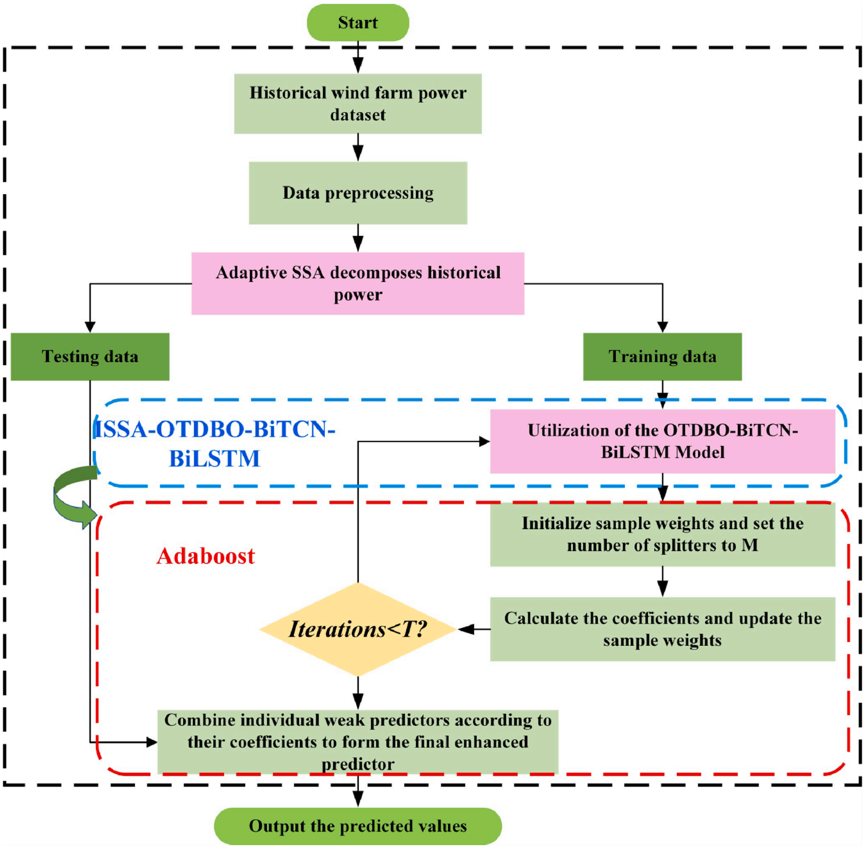

Start
Historical wind farm power dataset
Data preprocessing
Adaptive SSA decomposes historical power
Testing data
Training data
ISSA-OTDBO-BiTCN-BiLSTM
Utilization of the OTDBO-BiTCN-BiLSTM Model
Adaboost
Initialize sample weights and set the number of splitters to M
Iterations<T?
Calculate the coefficients and update the sample weights
Combine individual weak predictors according to their coefficients to form the final enhanced predictor
Output the predicted values

Figure 24. Workflow diagram of the Adaboost model

The model processes historical wind power data using adaptive SSA, then trains a BiTCN-BiLSTM predictor optimized by OTDBO. Adaboost iteratively updates sample weights and combines weak predictors into a final enhanced predictor for forecasting.

36. Lv, Y., Hu, Q., Xu, H., Lin, H., and Wu, Y. (2024). An ultra-short-term wind power prediction method based on spatial-temporal attention graph convolutional model. Energy *293*, 130751. https://doi.org/10.1016/j.energy.2024.130751.

37. Radwan, M., Alhussan, A.A., Ibrahim, A., and Tawfeek, S.M. (2025). Potato leaf disease classification using optimized machine learning models and feature selection techniques. Potato Res. *68*, 1–25. https://doi.org/10.1007/s11540-024-09763-8.

38. Wang, J., An, Y., Li, Z., and Lu, H. (2022). A novel combined forecasting model based on neural networks, deep learning approaches, and multi-objective optimization for short-term wind speed forecasting. Energy *251*, 123960. https://doi.org/10.1016/j.energy.2022.123960.

39. Liu, H., Tian, H.Q., and Li, Y.F. (2015). Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms. Energy Convers. Manag. *100*, 16–22. https://doi.org/10.1016/j.enconman.2015.04.057.

40. Abedinia, O., Lotfi, M., Bagheri, M., Sobhani, B., Shafie-Khah, M., and Catalão, J.P.S. (2020). Improved EMD-based complex prediction model for wind power forecasting. IEEE Trans. Sustain. Energy *11*, 2790–2802. https://doi.org/10.1109/TSTE.2020.2976038.

41. Huang, Y., Yang, L., Liu, S., and Wang, G. (2019). Multi-step wind speed forecasting based on ensemble empirical mode decomposition, long short term memory network and error correction strategy. Energies *12*, 1822. https://doi.org/10.3390/en12101822.

42. Ding, Y., Chen, Z., Zhang, H., Wang, X., and Guo, Y. (2022). A short-term wind power prediction model based on CEEMD and WOA-KELM. Renew. Energy *189*, 188–198. https://doi.org/10.1016/j.renene.2022.02.108.

43. Sun, Z., Zhao, S., and Zhang, J. (2019). Short-term wind power forecasting on multiple scales using VMD decomposition, K-means clustering and LSTM principal computing. IEEE Access *7*, 166917–166929. https://doi.org/10.1109/ACCESS.2019.2953580.

44. Karijadi, I., Chou, S.Y., and Dewabharata, A. (2023). Wind power forecasting based on hybrid CEEMDAN-EWT deep learning method. Renew. Energy *218*, 119357. https://doi.org/10.1016/j.renene.2023.119357.

45. Hou, G., Wang, J., and Fan, Y. (2024). Multistep short-term wind power forecasting model based on secondary decomposition, the kernel principal component analysis, an enhanced arithmetic optimization algorithm, and error correction. Energy *286*, 129640. https://doi.org/10.1016/j.energy.2023.129640.

46. Li, J., Zhang, S., and Yang, Z. (2022). A wind power forecasting method based on optimized decomposition prediction and error correction. Elec. Power Syst. Res. *208*, 107886. https://doi.org/10.1016/j.epsr.2022.107886.

47. Yang, J. (2019). A novel short-term multi-input–multi-output prediction model of wind speed and wind power with LSSVM based on improved ant colony algorithm optimization. Clust. Comput. *22*, 3293–3300. https://doi.org/10.1016/j.chaos.2019.03.003.

48. Huang, H., Jia, R., Shi, X., Liang, J., and Dang, J. (2021). Feature selection and hyperparameters optimization for short-term wind power forecast. Appl. Intell. *51*, 6752–6770. https://doi.org/10.1007/s10489-021-02191-y.

49. Doke, P., Shrivastava, D., Pan, C., Zhou, Q., and Zhang, Y.D. (2020). Using CNN with Bayesian optimization to identify cerebral micro-bleeds. Mach. Vis. Appl. *31*, 36. https://doi.org/10.1007/s00138-020-01087-0.

50. Tahir, M.F., Yousaf, M.Z., Tzes, A., El Moursi, M.S., and El-Fouly, T.H.M. (2024). Enhanced solar photovoltaic power prediction using diverse machine learning algorithms with hyperparameter optimization. Renew. Sustain. Energy Rev. *200*, 114581. https://doi.org/10.1016/j.rser.2024.114581.

51. Yang, L., and Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061.

52. Marini, F., and Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. Chemometr. Intell. Lab. Syst. *149*, 153–165. https://doi.org/10.1016/j.chemolab.2015.08.020.

53. Mirjalili, S. (2019). Genetic Algorithm. In: Evolutionary Algorithms and Neural Networks. In Studies in Computational Intelligence, *780* (Springer), pp. 43–55. https://doi.org/10.1007/978-3-319-93025-1_4.

54. Huang, C.L., and Dun, J.F. (2008). A distributed PSO–SVM hybrid system with feature selection and parameter optimization. Appl. Soft Comput. *8*, 1381–1391. https://doi.org/10.1016/j.asoc.2007.10.007.

55. Wang, S., Zhang, N., Wu, L., and Wang, Y. (2016). Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method. Renew. Energy *94*, 629–636. https://doi.org/10.1016/j.renene.2016.03.103.

56. El-kenawy, E.S.M., Khodadadi, N., Mirjalili, S., Abdelhamid, A.A., Eid, M. M., and Ibrahim, A. (2024). Greylag Goose Optimization: Nature-inspired optimization algorithm. Expert Syst. Appl. *238*, 122147. https://doi.org/10.1016/j.eswa.2023.122147.

57. Wan, A., Peng, S., AL-Bukhaiti, K., Ji, Y., Ma, S., Yao, F., and Ao, L. (2024). A novel hybrid BWO-BiLSTM-ATT framework for accurate offshore wind power prediction. Ocean Eng *312*, 119227. https://doi.org/10.1016/j.oceaneng.2024.119227.

58. Zhou, M., Wang, L., Hu, F., Zhu, Z., Zhang, Q., Kong, W., Zhou, G., Wu, C., and Cui, E. (2024). ISSA-LSTM: A new data-driven method of heat load forecasting for building air conditioning. Energy Build. *321*, 114698. https://doi.org/10.1016/j.enbuild.2024.114698.

59. Quan, R., Qiu, Z., Wan, H., Yang, Z., and Li, X. (2024). Dung beetle optimization algorithm-based hybrid deep learning model for ultra-short-term PV power prediction. iScience *27*, 111126. https://doi.org/10.1016/j.isci.2024.111126.

60. Geng, D., Zhang, Y., Zhang, Y., Qu, X., and Li, L. (2025). A hybrid model based on CapSA-VMD-ResNet-GRU-attention mechanism for ultra-short-term and short-term wind speed prediction. Renew. Energy *240*, 122191. https://doi.org/10.1016/j.renene.2024.122191.

61. Wu, B., Wang, L., and Zeng, Y.R. (2022). Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. Energy *252*, 123990. https://doi.org/10.1016/j.energy.2022.123990.

62. Wang, S., Shi, J., Yang, W., and Yin, Q. (2024). High and low frequency wind power prediction based on Transformer and BiGRU-Attention. Energy *288*, 129753. https://doi.org/10.1016/j.energy.2023.129753.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Wind power dataset (2022, Dabancheng Wind Farm)	This paper	Available upon request from the corresponding author
Software and algorithms		
MATLAB 2023b	MathWorks	R2023b
LSTM	Liu, X et al.[33]	https://doi.org/10.1016/j.asoc.2023.111050
CNN	Lawal, A et al.[35]	https://doi.org/10.1109/ACCESS.2021.3129883
DBO	Quan, R et al.[59]	https://doi.org/10.1016/j.isci.2024.111126
WOA	Wan, A et al.[57]	https://doi.org/10.1016/j.oceaneng.2024.119227
Code for development and evaluation	This paper	https://doi.org/10.7910/DVN/4JXZV8

METHOD DETAILS

Adaptive singular spectrum analysis mode decomposition based on autocorrelation function

Traditional Singular Spectrum Analysis (SSA) relies on empirical selection or trial-and-error to determine decomposition components, introducing subjective bias and reducing flexibility in handling complex wind power sequences. To address this, this study proposes Adaptive SSA (ISSA), which automatically determines the optimal number of subsequences using data autocorrelation, eliminating manual parameter tuning. This approach enhances decomposition accuracy, improves model adaptability, and ensures more reliable wind power forecasts. The working steps of ISSA are as follows:

Matrix decomposition

In the embedding step, a suitable window length L (2<L<N/2) is required first, and the one-dimensional wind power time series is arranged with lag to construct the trajectory matrix X.

$$\boldsymbol{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix} \qquad \text{(Equation 1)}$$

Where K=N-L+1. The trajectory matrix X is an L×K Hankel matrix, and each column is a subsequence of the time series.

Decomposition

In the decomposition step, the covariance matrix S of the trajectory matrix is first calculated, and then the eigenvalue decomposition of S is performed to obtain the eigenvalue $\lambda_1 > \lambda_2 > \cdots > \lambda_L \geq 0$ and the eigenvector $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_L]$, and the eigenvalue λ_i is the singular spectrum of the original sequence, and there are:

$$\mathbf{X} = \sum_{m=1}^{L} \sqrt{\lambda_m} \mathbf{U}_m \mathbf{V}_m^T, \mathbf{V}_m = \frac{\mathbf{X}^T \mathbf{U}_m}{\sqrt{\lambda_m}}, m = 1, 2, \ldots, L \qquad \text{(Equation 2)}$$

Among them, the eigenvector $\mathbf{U}_i$ corresponding to λ_i reflects the evolution type of time series, which is called the time empirical orthogonal function.

Grouping

In grouping, set the subscript {1,2, …, L} partition into M disjoint subsets $I_1, I_2, \ldots I_M$. Let $I = \{i_1, i_2, \ldots, i_p\}$, then the composite matrix corresponding to I is:

$$\mathbf{X}_I = \mathbf{X}_{i_1} + \mathbf{X}_{i_2} + \cdots + \mathbf{X}_{i_p} \qquad \text{(Equation 3)}$$

$$\mathbf{X} = \mathbf{X}_{I_1} + \mathbf{X}_{I_2} + \cdots + \mathbf{X}_{I_M} \qquad \text{(Equation 4)}$$

Reconstruction

The grouping matrix $X_l=(y_{ij})_{L\times K}$ is converted to the corresponding sequence $RC_l=(rc_1,rc_2,\ldots r_{CN})$. The k-th element in the RC_l is the mean of all elements in the matrix X_l that satisfy i+j=k+1. The specific steps are as follows:

$$rc_k = \begin{cases} \frac{1}{k}\sum_{m=1}^{k} y_{m,k-m+1}, 1 \le k<L \\ \frac{1}{L}\sum_{m=1}^{L} y_{m,k-m+1}, L \le k<K \\ \frac{1}{N-k+1}\sum_{m=k-K+1}^{N-K+1} y_{m,k-m+1}, K \le k \le N \end{cases} \tag{Equation 5}$$

The trajectory matrix of each group is reconstructed into a time series, and the reconstruction time series of all groups are added together to get the final reconstruction time series.

Adaptive dimension selection

In traditional SSA, the window size (embedding dimension) L is manually selected, potentially leading to under- or over-decomposition of the time series. To eliminate subjectivity, this study employs an autocorrelation function (ACF)-based approach for automatic L determination. Let the time series be $\{x_t\}_t^N$, where N is the length of the time series. First, calculate the autocorrelation function (ACF) of the time series:

$$\rho(k) = \frac{\sum_{t=1}^{N-k}(x_t - \overline{x})(x_{t+k} - \overline{x})}{\sum_{t=1}^{N}(x_t - \overline{x})^2} \tag{Equation 6}$$

Where, $\overline{x}$ is the mean of the time series, and k is the lag value. Next, set a threshold θ and find the first lag value that makes the autocorrelation function $\rho(k)$ less than the threshold θ:

$$M = \min\{k|\rho(k) < \theta\} \tag{Equation 7}$$

The lag value M is selected as the adaptive embedding dimension L. This adaptive selection method can automatically determine the decomposition dimension according to the statistical characteristics of the time series, thereby improving the accuracy of SSA decomposition.

Dung beetle optimization algorithm

This study proposes an enhanced DBO algorithm to automate hyperparameter optimization for BiTCN-BiLSTM, improving wind power forecasting accuracy. The BiTCN-BiLSTM model captures spatiotemporal features, but its performance hinges on fine-tuning key hyperparameters (e.g., BiLSTM neurons, BiTCN filters, and learning rate). Traditional manual tuning struggles with balancing global exploration and local optimization.

The improved DBO algorithm leverages rolling and dancing for local search, foraging and stealing for global exploration, and a reproduction mechanism to prevent premature convergence. By integrating population segmentation and boundary selection, it boosts search efficiency, accelerates optimization, and enhances forecasting performance.

Rolling dung beetles

The dung beetle turns the dung into a ball and rolls it into the desired position. The position update formula of the rolling dung beetle can be expressed as:

$$\begin{aligned} &x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \\ &\Delta x = |x_i(t) - X^w| \end{aligned} \tag{Equation 8}$$

Where t denotes the current iteration, and x_i (t) represents the position of the i-th dung beetle at iteration t. The parameter k ∈ (0,0.2) is the deflection coefficient, while b ∈ (0,1) is a constant. The natural coefficient α is set to -1 or 1, where 1 indicates no deviation, and -1 signifies a directional shift. X^w denotes the global worst position, and Δx simulates changes in light intensity. When a ball-rolling dung beetle encounters an obstacle, it adjusts direction using dancing behavior. The position update equation for dancing behavior is as follows

$$x_i(t+1) = x_i(t) + \tan(\theta)|x_i(t) - x_i(t-1)| \tag{Equation 9}$$

Where θ∈[0,π], if θ is equal to 0 or π/2, the position of the dung beetle will not be updated.

Breeding dung beetles

The female dung beetle rolls the dung ball to a safe area for egg-laying, modeled using a boundary selection strategy to simulate the spawning zone. The strategy is:

$$\begin{aligned} Lb^{*} &= \max(X^{*} \times (1 - R), Lb), \\ Ub^{*} &= \min(X^{*} \times (1 + R), Ub) \end{aligned} \quad \text{(Equation 10)}$$

Where X* denotes the current local optimal position, while Lb* and Ub* define the spawning area's bounds. R = 1 − t/Tmax, where T_{max} is the maximum iteration count, and Lb, Ub represent the optimization problem's bounds. Each female dung beetle lays one egg per iteration within the spawning area. The hatching ball's position remains dynamic throughout iterations, and its update process is as follows:

$$B_i(t + 1) = X^{*} + b_1 \times (B_i(t) - Lb^{*}) + b_2 \times (B_i(t) - Ub^{*}) \quad \text{(Equation 11)}$$

Where B_i (t) represents the position of the i-th hatching ball at iteration t. b_1 and b_2 are two independent 1×D random vectors, where D is the dimension of the optimization problem. The hatching ball's position is strictly confined to the spawning area.

Small dung beetle (foraging dung beetle)

Successfully hatched eggs develop into juvenile dung beetles, which forage within a designated optimal foraging area. The boundaries of this area are categorized as follows:

$$\begin{aligned} Lb^{b} &= \max\left(X^{b} \times (1 - R), Lb\right), \\ Ub^{b} &= \min\left(X^{b} \times (1 + R), Ub\right) \end{aligned} \quad \text{(Equation 12)}$$

Where X^b represents the global optimal position, $Lb^b$and Ub^b represent the lower and upper bounds of the optimal foraging area, respectively, and other parameters are defined in Equation 11. Therefore, the position of the dung beetle is updated as follows:

$$x_i(t + 1) = x_i(t) + C_1 \times \left(x_i(t) - Lb^{b}\right) + C_2 \times \left(x_i(t) - Ub^{b}\right) \quad \text{(Equation 13)}$$

Where x_i (t) represents the position information of the i-th dung beetle at the t-th iteration, C_1 represents a random number following the normal distribution, and C_2 represents a random vector belonging to (0,1).

Stealing dung beetle

Some dung beetles will steal dung balls from other dung beetles. During the iteration, the location information of the thieves is updated, which can be described as follows:

$$x_i(t + 1) = X^{b} + S \times g \times \left(|x_i(t) - X^{*}| + \left|x_i(t) - X^{b}\right|\right) \quad \text{(Equation 14)}$$

Where x_i (t) is the position information of the i-th thieving dung beetle at the t-th iteration, g is a random vector of size 1×D that obeys the normal distribution, and S is a constant value.

Improved dung beetle optimization algorithm

Population initialization based on Kent mapping

The original DBO algorithm relies on random initialization, often leading to uneven population distribution and reduced search efficiency. Compared to random initialization, Latin hypercube sampling, and Sobol sequences, chaotic mapping offers better ergodicity and sensitivity to initial values, generating diverse populations and enhancing global search capability.

To improve solution space exploration, this study integrates Kent mapping from chaos theory for enhanced population initialization. Assuming n dung beetle individuals in a d-dimensional space, the process begins by randomly generating a d-dimensional vector x_1 (first dung beetle) within [0,1]. The remaining n-1 vectors are then iteratively generated using Kent chaotic mapping, defined by the following equation:

$$x_{n+1} = \begin{cases} \dfrac{x_n}{a} & 0 < x_n \leq a \\ \dfrac{1 - x_n}{(1 - a)} & a < x_n < 1 \end{cases} \quad \text{(Equation 15)}$$

Where, *a* is the control parameter. When *a* = 0.5, the system will present a short-period state. Therefore, the value range of in this paper is (0.01, 0.5). x_{n+1} is the position of the dung beetle individual after chaotic mapping. The value after chaotic mapping is mapped to the search space of the solution. The formula is as follows:

$$X_i = L + chaos \times (U - L) \quad \text{(Equation 16)}$$

Where L and U are the search space bounds, X_i is the initial position, and chaos is the chaotic factor generated by the Kent function.

Figure 18 compares Kent and Logistic mappings. Logistic mapping creates search blind spots, reducing efficiency, while Kent mapping ensures uniform initialization with superior ergodicity.

Combined with Osprey optimization algorithm(OOA)

The original DBO algorithm relies only on the worst solution, lacks population information exchange, and requires numerous control parameters. The Osprey Optimization Algorithm (OOA), inspired by ospreys' visual predation, enhances global exploration and avoids local optima through large-scale position updates.

To reduce DBO's dependence on the worst solution, this study integrates OOA's global search strategy into the rolling phase, replacing the original position update Equation 8. The osprey strategy detects the dung ball's position and executes rolling movements, enabling comprehensive solution space exploration. The OOA's first-phase global exploration strategy is formulated as follows:

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \cdot \left(SF_{i,j} - I_{i,j} \cdot x_{i,j}\right) \qquad \text{(Equation 17)}$$

$$x_{i,j}^{P1} = \begin{cases} x_{i,j}^{P_1}, lb_j \leq x_{i,j}^{P_1} \leq ub_j \\ lb_j, x_{i,j}^{P_1} < lb_j \\ ub_j, x_{i,j}^{P_1} > ub_j \end{cases} \qquad \text{(Equation 18)}$$

Where $x_{i,j}$ is an individual, lb_j is the lower boundary of the optimal solution, ub_j is the upper boundary of the optimal solution, SF is the fish selected by the osprey, r is a random number between [0,1], and the value of I is one of {1,2}.

Dynamic spiral search strategy

In the original DBO's breeding phase, female dung beetles reproduce within a spawning area, accelerating population convergence but reducing diversity, increasing the risk of local optima. The Whale Optimization Algorithm (WOA) employs a spiral search strategy, iteratively updating positions to enhance diversity while maintaining convergence speed. To improve DBO's population diversity, this study integrates a dynamic spiral search strategy into its position update mechanism. The spiral bubble method's position update formula in WOA is as follows:

$$\begin{aligned} X(t+1) &= D' \cdot e^{cl} \cdot \cos(2\pi l) + X^*(t) \\ D' &= |X^*(t) - X(t)| \end{aligned} \qquad \text{(Equation 19)}$$

Where, D' represents the distance between the current individual and the optimal position (food), l is a random number in [-1,1], and c is a constant defining the spiral shape. To prevent rapid decay or slow convergence, this study modifies c from a static to a dynamic spiral search factor. The calculation equation for the dynamic spiral search factor r is:

$$r = e^{k \times \cos\left(\frac{\pi \cdot t}{Max\_teration}\right)} \qquad \text{(Equation 20)}$$

Where r varies with iterations, dynamically adjusting the spiral's size and amplitude based on a cosine function. k is the change coefficient, set to 2. With the dynamic spiral search strategy, the dung beetle's search path in the breeding stage is illustrated in Figure 19, and the DBO position update formula is modified as follows:

$$B_i(t+1) = X^* + e^{rl} \cdot \cos(2\pi l) \times b_1 \times \left(B_i(t) - Lb^*\right) + e^{rl} \cdot \cos(2\pi l) \times b_2 \times \left(B_i(t) - Ub^*\right) \qquad \text{(Equation 21)}$$

Adaptive t-distribution perturbation strategy

During foraging, smaller dung beetles remain confined to the optimal zone, limiting exploration. To overcome this, an adaptive t-distribution perturbation strategy enhances global search in early stages and local refinement later. The t-distribution's heavy tails increase large perturbation probability, aiding escape from local optima.

The updated positioning method for foraging dung beetles incorporating this strategy is:

$$x_i(t+1) = \begin{cases} x_i(t) + C_1 \times \left(x_i(t) - Lb^b\right) + C_2 \times \left(x_i(t) - Ub^b\right) \\ X_{\text{best}} + \text{trnd}(\nu) \cdot X_{\text{best}} \end{cases} \qquad \text{(Equation 22)}$$

Where trnd (ν) generates a random number following a t-distribution with degrees of freedom.

The adaptive t-distribution's degrees of freedom is calculated as follows:

$$\nu = \exp\left(4\left(\frac{t}{M}\right)^2\right) \qquad \text{(Equation 23)}$$

Where t represents the current iteration number, M represents the maximum iteration number, and ν represents the degree of freedom parameter of t distribution. The complete OTDBO algorithm flow is shown in Figure 20.

Bidirectional temporal convolutional network(BiTCN)

BiTCN extends TCN by incorporating bidirectional dilated causal convolution and residual blocks, enhancing long-term dependency modeling and stable training. Its architecture consists of two parallel TCNs—one for historical sequences and one for future covariates—expanding the receptive field and improving forecasting accuracy. Dilated convolution reduces layer depth while maintaining computational efficiency, mitigating gradient vanishing (Figure 21A). Meanwhile, residual blocks enable identity mapping, stabilizing deep network training and improving error propagation, preventing both gradient explosion and vanishing. Stacking multiple BiTCN modules further strengthens feature extraction and enhances complex time-series pattern recognition, making it particularly effective for wind power forecasting (Figure 21B). More specifically, the forward dilated convolution operation f for elements s and filters f∈R with sequential input x∈R can be defined as follows:

$$F(s) = \sum_{j=0}^{k-1} f(j)\cdot \mathbf{x}_{s+d\cdot j}, \quad \text{(Equation 24)}$$

Where k is the filter size, d is the expansion factor, controlling the spacing between convolutional kernels. The dilation rate grows exponentially across layers, expanding the receptive field. The forward step is s+d·j, while for backward causal convolution, it modifies to s-d·j in Equation 24.

Bidirectional long short-term memory network(BiLSTM)

BiLSTM enhances temporal feature extraction through bidirectional information flow, enabling a more comprehensive capture of short-term fluctuations and long-term trends in wind power data, thereby improving prediction accuracy.

As shown in Figure 22, BiLSTM consists of forward and backward LSTM layers, which work in parallel to process sequential data from both directions. Combined with a gating mechanism, BiLSTM selectively retains key information, effectively modeling long-term dependencies while ensuring stable and accurate predictions.

The calculation process of LSTM is as follows:

(1) Calculate the forget gate and select the information to be forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(Equation 25)}$$

Where σ is the sigmoid activation function, W_f is the weight matrix of the forgetting gate, b_f is the bias term, h_{t-1} is the hidden state of the previous time step, and x_t is the input of the current time step.

(2) Calculate the input gate

$$\begin{aligned} i_t &= \sigma(W_i\cdot[h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C\cdot[h_{t-1}, x_t] + b_C) \end{aligned} \quad \text{(Equation 26)}$$

Where i_t represents the output of the input gate, W_i and W_C are the weight matrix, $\tilde{C}_t$ the candidate memory unit, and b_i and b_C are the bias terms.

(3) Update LSTM cell status

$$C_t = f_t\cdot C_{t-1} + i_t\cdot\tilde{C}_t \quad \text{(Equation 27)}$$

Where C_{t-1} is the cell state of the previous time step.

(4) Calculation output gate

$$\begin{aligned} o_t &= \sigma(W_o\cdot[h_{t-1}, x_t] + b_o) \\ h_t &= o_t\cdot\tanh(C_t) \end{aligned} \quad \text{(Equation 28)}$$

Where o_t is the output of the output gate, and h_t represents the updated hidden state.

This study proposes the BiTCN-BiLSTM model (Figure 23) to enhance spatiotemporal feature extraction in wind power forecasting. BiTCN leverages dilated convolution and residual connections to capture multi-scale and long-term dependencies while preserving bidirectional temporal information. Subsequently, BiLSTM models the sequential features extracted by BiTCN, learning the dynamic evolution patterns of the data. BiTCN's convolutional operations enable fast feature extraction, while BiLSTM's gating mechanism effectively captures long-term dependencies. This combination improves prediction accuracy while enhancing model robustness and generalization capability.

Adaptive boost (Adaboost) power prediction correction

To enhance stability and accuracy, this study integrates Adaboost ensemble learning into the OTDBO-BiTCN-BiLSTM model. Figure 24 illustrates the AdaBoost workflow. Adaboost iteratively trains multiple weak predictors (OTDBO-BiTCN-BiLSTM rounds), adjusting sample weights to emphasize hard-to-predict cases. By aggregating weighted predictions, it forms a strong predictor, correcting local biases and improving adaptability to wind power fluctuations. In this framework, OTDBO-BiTCN-BiLSTM captures nonlinear dependencies between meteorological features and power sequences, while Adaboost mitigates uncertainties through ensemble learning, ensuring more accurate and stable forecasts. The methodology is outlined as follows:

(1) Preliminary prediction: Use the OTDBO-BiTCN-BiLSTM model to make preliminary predictions for k wind power subsequences and obtain the initial prediction value $h_t(x_i)$.
(2) Error calculation: Calculate the error between the preliminary prediction value and the true value $\epsilon_t = y_t(i) - h_t(x_\mathrm{i})$.
(3) Adaboost Correction: Utilize the error sequence as the input for AdaBoost, train a series of BiTCN-BiLSTM weak learners, and construct a correction model. The weak learners are aggregated using the AdaBoost algorithm to produce a robust corrector, denoted as H(x). The process of training weak predictors in Adaboost is shown in Equation 30.

$$\begin{cases} D_t(i) = \frac{1}{N}, i = 1, 2, ..., N \\ e_t = \sum_{i=1}^{N} D_t(i) I \quad (y_t(i) \neq G_t(x_i)) \\ a_k = 0.5 \ln(1/e_t - 1) \\ D_{t+1}(i) = \frac{D_t(i) \exp(-a_t y_t(i) G_t(x_i))}{Z_t} \end{cases} \quad \text{(Equation 29)}$$

Where, $D_t(i)$ represents the weight of each sample after initialization; e_t represents the prediction error rate; I is the indicator function; a_t represents the weight of each predictor; $\mathrm{y_t(i)}$ represents the actual value; c is a constant; through iterative calculation, the weight value of the sample is continuously updated, and finally the weak predictors $\mathrm{h_t(x_t)}$ are combined to obtain a strong predictor.

$$H(x) = sign\left(\sum_{t=1}^{T} \mathrm{e}_t h_t(x_t)\right) \quad \text{(Equation 30)}$$

(4) Final Prediction: Integrate the preliminary prediction values of the wind power subsequences with the output of the corrector to derive the corrected subsequence predictions. Subsequently, the overall wind power prediction is derived by aggregating the corrected predictions from k subsequences.

$$h^{\mathrm{final}} = \sum_{k=1}^{k} \left(h_t(x_i) + sign\left(\sum_{t=1}^{T} \mathrm{e}_t h_t(x_t)\right)\right) \quad \text{(Equation 31)}$$

Workflow of the ISSA-OTDBO-BiTCN-BiLSTM-Ada model

To fully extract latent patterns from historical wind power data, this study first applies ISSA for modal decomposition, capturing long-term trends and short-term fluctuations across frequency domains. BiTCN-BiLSTM then models local and global dependencies, while OTDBO optimizes hyperparameters for efficient multi-scale feature learning and nonlinear mapping. Finally, Adaboost iteratively refines predictions, enhancing robustness and accuracy.

The complete forecasting process consists of:

(1) Data Preprocessing: Wind power data, collected at 15-minute intervals, undergoes outlier removal, missing value imputation, and normalization.
(2) Modal Decomposition & Feature Selection: ISSA decomposes data into sub-sequences, while a random forest algorithm ranks meteorological features (e.g., wind speed, temperature), selecting the most relevant ones for dataset partitioning.
(3) Hyperparameter Optimization: The optimal OTDBO-BiTCN-BiLSTM hyperparameters are determined by minimizing RMSE on the training set.
(4) Model Training & Prediction Refinement: The optimized BiTCN-BiLSTM generates sub-sequence predictions, which Adaboost iteratively refines by reweighting high-error samples, reducing local biases. The final wind power predictions are reconstructed from corrected sub-sequences, improving accuracy and stability.

Objective function and evaluation metrics

The objective function of OTDBO is the root mean square error (RMSE), which is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{n}} (\hat{\mathrm{y}}_\mathrm{i} - \mathrm{y_i})^2} \quad \text{(Equation 32)}$$

Where, $\hat{y}_\mathrm{i}$ is the predicted value of wind power, y_i is the measured value of wind power, and n is the number of samples.

Selecting appropriate evaluation metrics is crucial for wind power forecasting. While MAPE is widely used, it becomes unreliable when actual values are zero and is highly sensitive to small values, potentially exaggerating errors.

To ensure robust evaluation, this study adopts MAE, RMSE, and R^2, which effectively handle zero values while providing a comprehensive performance assessment. Additionally, three supplementary metrics—IR$_{\mathrm{RMSE}}$, IR$_{\mathrm{MAE}}$, and IR$_{\mathrm{R}}{}^2$—are introduced to quantify performance improvements across models.

The calculation formula is as follows:

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}} \left|\hat{\mathrm{y}}_i - \mathrm{y}_i\right| \qquad \text{(Equation 33)}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{n}} \left(\hat{\mathrm{y}}_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}}\right)^2} \qquad \text{(Equation 34)}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (\hat{y}_i - \overline{y})^2} \qquad \text{(Equation 35)}$$

$$IR_{\mathrm{MAE}} = \frac{\mathrm{MAE}_{\mathrm{compared}} - \mathrm{MAE}_{\mathrm{proposed}}}{\mathrm{MAE}_{\mathrm{compared}}} \qquad \text{(Equation 36)}$$

$$IR_{\mathrm{RMSE}} = \frac{\mathrm{RMSE}_{\mathrm{compared}} - \mathrm{RMSE}_{\mathrm{proposed}}}{\mathrm{RMSE}_{\mathrm{compared}}} \qquad \text{(Equation 37)}$$

$$IR_{R^2} = \frac{R^2_{\mathrm{proposed}} - R^2_{\mathrm{compared}}}{R^2_{\mathrm{proposed}}} \qquad \text{(Equation 38)}$$

In Equations 33 and 34, n represents the number of samples; $\overline{y}$ represents the average value of the actual power.

QUANTIFICATION AND STATISTICAL ANALYSIS

All model evaluations and statistical analyses in this study were performed using MATLAB R2023b software, and graphical visualizations were generated using Origin 2021 software. To comprehensively assess model performance, the following statistical metrics were employed: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). Calculation formulas and definitions of these metrics are provided in Equations 33, 34, and 35. Furthermore, Improvement Rate (IR) metrics—including IR$_{\mathrm{RMSE}}$, IR$_{\mathrm{MAE}}$, and IR$_{\mathrm{R}}{}^2$—were introduced to quantify performance enhancements among different models, with their calculations detailed in Equations 36, 37, and 38.

To statistically assess the significance of prediction accuracy improvements between the proposed model and benchmark models, a one-tailed paired t-test was conducted across four seasonal datasets (March, June, September, and December) based on MAE, RMSE, and R^2 metrics. The results of the significance tests, including the corresponding p-values, are presented in Table 14. Specifically, the null hypothesis (H$_0$) states that there is no significant difference in prediction accuracy between the proposed model and the benchmarks, whereas the alternative hypothesis (H$_1$) asserts that the proposed model significantly outperforms the benchmarks, with a significance level (α) set at 0.05.

The sample size n represents the number of test samples for each month's dataset, with n = 595 for March and December and n = 576 for June and September. All evaluation metrics (RMSE, MAE, R^2, and IR metrics) were calculated using the entire testing datasets to represent the overall predictive capability of the models.